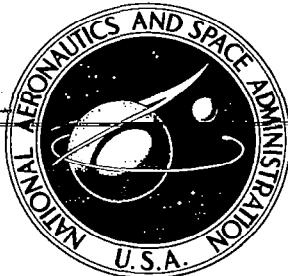


NASA CONTRACTOR  
REPORT



NASA CR-2

0061682



TECH LIBRARY KAFB, NM

LOAN COPY: RETURN TO  
AFWL TECHNICAL LIBRARY  
KIRTLAND AFB, N. M.

NASA CR-2821

TRANDES: A FORTRAN PROGRAM  
FOR TRANSONIC AIRFOIL  
ANALYSIS OR DESIGN

*Leland A. Carlson*

*Prepared by*

TEXAS A&M UNIVERSITY

College Station, Tex. 77843

*for Langley Research Center*

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JUNE 1977



0061682

1. Report No. NASA CR-2821	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle TRANDES: A Fortran Program for Transonic Airfoil Analysis or Design		5. Report Date June 1977	6. Performing Organization Code
7. Author(s) Leland A. Carlson		8. Performing Organization Report No.	
9. Performing Organization Name and Address Texas A&M University College Station, Texas 77843		10. Work Unit No. 505-06-33-08-00	11. Contract or Grant No. NSG-1174
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Contractor Report	
14. Sponsoring Agency Code			
15. Supplementary Notes Final report Langley Technical Monitor: Jerry C. South, Jr.			
16. Abstract A program called TRANDES is presented that can be used for the analysis of steady, irrotational, transonic flow over specified two-dimensional airfoils in free air or for the design of airfoils having a prescribed pressure distribution, including the effects of weak viscous interaction (i.e., massive separation is not considered). Instructions on program usage, listings of the program, and sample cases are given.			
17. Key Words (Suggested by Author(s)) Supercritical airfoil design Viscous interaction Relaxation Cartesian coordinates		18. Distribution Statement Unclassified - Unlimited  Subject Category 02	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 109	22. Price* \$5.50

TRANDES: A FORTRAN PROGRAM FOR TRANSONIC  
AIRFOIL ANALYSIS OR DESIGN

By  
Leland A. Carlson  
Texas A&M University

SUMMARY

A program called TRANDES is presented that can be used for the analysis of steady, irrotational, transonic flow over specified two-dimensional airfoils in free air or for the design of airfoils having a prescribed pressure distribution, including the effects of weak (no massive separation) viscous interaction. Instructions on program usage, listings of the program, and sample cases are given.

INTRODUCTION

The program described in this report can be used for either the analysis of the flowfield about an airfoil in free air or for the design of an airfoil having a specified pressure distribution. In the direct or analysis mode the airfoil shape is prescribed and the flowfield and surface pressures are determined. In the direct-inverse or design mode an initial nose shape is given along with the pressure distribution on the remainder of the airfoil, and the flowfield and actual airfoil shape are computed. In either case, the effects of weak viscous interaction may be included at the option of the user.

The program solves the exact equation for the perturbation potential in a Cartesian coordinate system. Most of the background about the equations solved, formulation of the boundary conditions, and the difference scheme used is given in references 1-3. This report gives instructions on the use of the computer program and also some additional details concerning the inclusion of weak viscous interaction. It should be noted that in this report the term weak viscous interaction implies that there is no massive boundary layer separation on the airfoil. Nevertheless, for aft-loaded airfoils at transonic speeds, the effect of viscous interaction on airfoil performance may still be quite large.

The next section gives a general description of the problem and the method of solution. Then the instructions for using the computer program and a description of the inputs and outputs are given. The appendices contain

additional details as well as listings of the program and the sample cases.

#### GENERAL DESCRIPTION

The program described in this paper obtains the inviscid flowfield by solving the full, inviscid, perturbation-potential flow equation in a cartesian grid system. This system which is usually aligned relative to the airfoil chord line, has been found to efficiently yield accurate solutions for biconvex, conventional, and aft-cambered airfoils. In the program, a stretching is applied to the coordinates such that the infinite physical plane is mapped to a finite computational space. Thus, the boundary conditions at infinity can be applied directly and there is no need for an asymptotic far-field solution. Details about the stretching functions are given in appendix A.

The method of solution is to replace the governing second-order partial differential equation with a non-conservative system of finite difference equations that includes, at supersonic points, a form of Jameson's "rotated" difference scheme (ref.4). The difference equations are then solved by column relaxation, which in order to obtain rapid convergence is usually done on several different grids. In the analysis case, the difference equations are first solved on a very coarse grid (typically 13x7). The solution is then interpolated and used as an initial condition for a coarse grid (typically 25x13). This procedure can be repeated twice more to obtain solutions on a medium grid (49x25) and on a fine grid (97X49). The latter has 130 points on the airfoil; however, excellent results are usually obtained on the medium grid, particularly considering the computer time involved. For typical examples see reference 1.

In the inverse case, which is normally used for airfoil modification or design, an initial airfoil shape must be assumed. However, this choice is not critical, and the final airfoil shape may be considerably different. Since experience indicates that the inverse scheme works best if the perturbation potentials have reasonable initial values, fifty relaxation cycles are first performed in the direct mode for the initial airfoil shape on a very coarse grid. The grid is then halved and the inverse procedure initiated using the input pressure distribution as the boundary condition in the inverse region. As in the direct case, the grid may be refined again to the medium grid (typically 49x25) where the results are usually adequate. Fine grid usage in the inverse

case is not recommended due to slow convergence. On each grid the airfoil shape is recomputed every ten relaxation cycles after the first fifty.

After the solution has been obtained in the design case, the resultant shpae is treated as a displacement surface and the displacement thickness is automatically subtracted to obtain the actual airfoil surface. The boundary layer characteristics are determined by the Nash-Macdonald method (ref.5) with smoothing.

The effects of viscous interaction may also be included in analysis cases at the option of the user. To preserve numerical consistency with the inverse scheme, the Nash-Macdonald method is also used in such analysis cases, starting with the 50th cycle on the coarse grid. At that point the displacement thickness is computed at the same x coordinates as the inviscid grid and the displacement surface ordinates updated using under-relaxation. The slopes are then determined from cubic splines through the new ordinates, which are updated by a new boundary layer calculation every ten relaxation cycles thereafter. For those cases having extensive trailing edge separation an empirical boundary layer correction is available. However, it is not necessary for most cases.

It should be noted that while the program can include the effects of boundary layer interaction, no correction has been applied for the effects of wake curvature and an empirical approach has been used in the trailing edge region. Thus, the results should be viewed as pressure versus lift coefficient, moment vs. lift, etc. instead of angle of attack. However, the error in angle of attack is believed to be small.

Typical total computation times on an Amdahl 470/V6 or a CYBER 175-T are 60-70 seconds for medium grid results and less than 250 seconds for fine grid solutions.

#### PROGRAM USAGE

The program is written in FORTRAN IV programming language for use on IBM 360-370, Amdahl 470, CDC 6600, and CDC CYBER series computers. The program can be overlaid in order to reduce computer storage, if required. In nonoverlay mode it requires less than 200,000 bytes on an IBM type machine. Some modification to formats etc. may be required to run the program on different computer systems.

The input cards are summarized in the following table:

Read Order	Variables	Format
1	NTITLE	20A4
2	NAMELIST/FINP/M,W,X1,X2,ALP,EPS,EPSS, X4,S4,CONV,A1,A2,A3,RN,XIBDLY,CIR,CDCORR, RDEL,RDELFN,SP,XSEP,XLSEP,XPC	Namelist
3	NAMELIST/IINP/IMAX,JMAX,IKASE,INV,MITER, NHALF,ITACT,ISKP2,ISKP3,ISKP4,ITERP,IREAD, LP,ITEUPC,ITELWC	Namelist
4	P(I,J) I=1,IMAX; J=1,JMAX (Only if IREAD=1)	5E15.7
5	PB(I) I=1,IMAX (Only if IREAD=1)	5E15.7
6*	X1,X2	2F10.5
7	NI	I5
8	XI(I),YI(I), I=1,NI	8F10.4
9	DERIX,DERIY,DERFX,DERFY	8F10.4
10	NIB	I5
11	XIB(I),YIB(I), I=1, NIB	8F10.4
12	DERIXB,DERIYB,DERFXB,DERFYB	8F10.4
13*	X1,X2	2F10.5
14*	CPU(I), I=I1, ITE	8F10.3
15*	CPL(I), I=I1, ITE	8F10.3
16*	X1,X2	2F10.5
17*	CPU(I), I=I1,ITE	8F10.3
18*	CPL(I), I=I1, ITE	8F10.3
19*	X1,X2	2F10.5
20*	CPU(I), I=I1, ITE	8F10.3

21\* CPL(I), I=I1, ITE

8F10.3

\* Read only in the design mode when INV=1

NOTE: In the design mode steps 13-15 are for the coarse grid, 16-18 for the medium and 19-21 for the fine grid (if used).

The definitions of these input variables are as follows:

- NTITLE - Description of case. Up to 80 alphanumeric characters. Appears on printed output, at the beginning of the results for each grid.
- M - Freestream Mach number (real variable). Default 0.5
- W - Relaxation factor for subsonic points. Should be in the range 0<W<2.0 Default 1.7
- X1 - X location where direct calculation stops. In analysis mode it should be set to 0.5 (i.e. trailing edge). In design mode it is usually set to slightly less than the third point from the leading edge or larger. Default 0.5
- X2 - End of the inverse region. For analysis case set to a large number. In inverse design case set to 0.5 (i.e. trailing edge). Default 10000.0
- ALP - Angle of attack in degrees. Default 0.0
- EPS - Subsonic damping factor to match difference equations at sonic line if needed. EPS has no effect on accuracy of solution, only on stability and convergence rate. Normally it is not needed. Default 0.0
- EPSS - Supersonic damping factor for iterative stability. Note that EPSS has no effect on the accuracy of the converged solution, only on the stability and convergence rate. EPSS should typically be about  $M_{max}^2 - 1$ , where  $M_{max}$  is the maximum local Mach number. Default 0.4
- X4 - The positive X location where the coordinate stretching changes. It should be near the airfoil trailing edge. Default 0.49.
- S4 - The positive  $\xi$  value in the computational plane where the stretching changes. Default 2.0
- CONV - Convergence criteria control value. Iterations stop when the maximum change in the perturbation potential (between relaxation cycles) is less than CONV. Default 1.E-05
- A1 - Stretching constant for the Y direction. It can be used to control  $\Delta Y$  and  $\Delta n$  near the horizontal axis. It is usually best to have  $\Delta \xi = \Delta n$

near the leading edge of the airfoil. Default 0.246

- A2 - First stretching constant for the X-direction. It is equivalent to  $\frac{2}{\pi} \left( \frac{dx}{d\xi} \right)$  at  $\xi = \xi_4$ . The value of  $A_2$  determines the horizontal step size near the leading and trailing edges, i.e.

$$\Delta x_{x=x_4} = \frac{\pi A_2}{2} \quad \Delta \xi = \frac{\pi A_2}{2} \frac{(2(1+S4))}{(IMAX-1)}$$

See Appendix A. Default 0.15

- A3 - Second stretching constant for the x-direction. It determines the physical location of the vertical grid line adjacent to grid side edge. Default 3.87.

- RN - Freestream Reynolds number based on chord length. Used only when viscous interaction included. Default 20.E+06.

- XIBDLY - The x-location at which transition is assumed to occur. The turbulent boundary layer calculation starts at the next grid point. The relationship to percent chord is

$$XIBDLY = (\% \text{ chord}-50.0)/100.0$$

Default -0.44.

- CIR - Circulation about airfoil. If an initial solution is inputted, it must be the corresponding value of circulation. ( $CIR = C_L/2.0$ ).

Default 0.0

- CDCORR - Correction to the computed wave drag coefficient for the finest grid used. Because of the lack of a large number of points in the leading and trailing edge regions, the wave drag coefficient has an error associated with grid size, spacing, and lift coefficient. The magnitude of CDCORR as a function of lift can be determined from a series of calculations at different angles of attack at subcritical speeds, where the wave drag should be zero. Note that the correction should be determined for each airfoil and grid combination. Default 0.0. See Appendix B.

- RDEL - Relaxation parameter for the boundary layer displacement thickness. It is used only when viscous interaction is included and  $IMAX \leq 55$ . Default 0.25

- RDELFN - Fine grid relaxation parameter for the boundary layer displacement thickness. It is used only when viscous interaction is included and

- IMAX > 55. Default 0.125
- SP - Maximum value allowed for the Nash-Macdonald separation parameter when  $x < XSEP$ . Used only in the viscous interaction case. Default 0.004.
- XSEP - X location after which the Nash-Macdonald separation parameter can assume its calculated value. Used only in the viscous interaction case. Default 0.44
- XLSEP - Location at which the trailing edge correction procedure begins. It should correspond to the point of separation, if used. Between XLSEP and the trailing edge the pressure distribution and the displacement surface is modified. Used only if ITEUPC and/or ITELWC equal 1. Default 0.50
- XPC - Location after which the lower surface displacement thickness is required to continue decreasing once it has started to decrease. Upstream of XPC the displacement thickness is required to be monotonically increasing. For most aft-cambered airfoils it should be 0.1 and in conventional airfoils it should be 0.5. Default 0.1
- IMAX - Number of vertical grid lines in the horizontal direction.  $I = 1$  is upstream infinity and  $I = IMAX$  is downstream infinity. For each grid refinement IMAX is increased such that  $IMAX_{new} = 2(IMAX_{old}) - 1$ . The limit on IMAX is 99. Default for use on first grid is 13.
- JMAX - Number of horizontal grid lines in the vertical direction.  $J = 1$  corresponds to infinity below the airfoil and  $J = JMAX$  is infinity above the airfoil. The same formula and limit that apply to IMAX also apply to JMAX. Default 7.
- IKASE - An integer number describing the case. It is limited to a maximum of six digits. Default 100.
- INV - Parameter determining program mode. It should be zero for analysis cases and one for inverse design cases. Default 0.
- MITER - Maximum number of iterations (complete relaxation cycles) allowed on first grid. MITER is halved for each grid refinement. However, on the fourth grid, MITER is reset to 400. Default 800.
- NHALF - Number of grid refinements to be done. Default 2.
- ITACT - Viscous interaction control parameter. It should be set to zero for analysis cases without interaction and for design cases. It should be one for analysis cases with interaction. Default 0.

- ISKP2** - Airfoil update control parameter for grid two. It should be 0 if on grid two an update is desired every 10 iterations. It should be 1 if an update is not desired until the grid two solution is completed.  
 Only used in the inverse design mode. Default 0.
- ISKP3** - Same as ISKP2 but for grid 3 (medium grid).
- ISKP4** - Same as ISKP2 but for grid 4 (fine grid).
- ITERP** - Interpolation parameter. If in the design mode the input  $C_p$  distribution for the grid 4 is to be read in, ITERP should be 0. If it is desired to linearly interpolate the  $C_p$  distribution of grid 3, it should be 1. Default 0.
- IREAD** - Starting solution control parameter. If IREAD is 0, the initial perturbation solution is assumed to everywhere be zero. If it is 1, an initial solution is read in from data cards. Default 0.
- LP** - Relaxation cycle interval at which boundary layer, surface ordinates, etc. details are printed. Useful for diagnostics. Default 1000.  
 (No printout.)
- ITEUPC** - Upper surface trailing edge correction control parameter. If trailing edge correction desired, ITEUPC should be 1. If not it should be zero.  
 Only used in the viscous interaction case. Normally the correction is not needed. Default 0.
- ITELWC** - Lower surface trailing edge correction control parameter. If correction desired, ITELWC should be 1. If not it should be 0. Only used in the viscous interaction case, and normally the correction is not needed. Default 0.
- P(I,J)** - Nondimensional perturbation potential,  $\phi_{ij}$ , at point I,J.
- PB(I)** - Nondimensional perturbation potential at point I on the  $y=0$ -grid line.
- X1, X2** - Same definition as above. However, in the inverse design case they must be read in prior to the solution of each grid. On the first grid (step 6 in above table) should use  $X1=0.5$ ,  $X2=10000.0$ . On remaining grids (steps 13,16, and 19), X1 should be the location where the direct calculation stops and X2 should be 0.5.
- NI** - The number of coordinate pairs used to describe the upper surface of the airfoil. Presently limited to 99.
- XI(I)** - Input coordinates in the horizontal direction for the airfoil upper

surface. The leading edge corresponds to  $XI=0.0$  and the trailing edge is  $XI=1.0$ .

**YI(I)** - Input coordinates in the vertical direction for the airfoil upper surface.

**DERIX** -  $DX/DS$  of the airfoil upper surface at the leading edge ( $XI=0.0$ ). It usually is 0.0.

**DERIY** -  $DY/DS$  of the airfoil upper surface at the leading edge ( $XI=0.0$ ). It usually is 1.0.

**DERFX** -  $D^3X/DS^3$  of the airfoil upper surface at the trailing edge ( $XI=1.0$ ). It is usually sufficiently accurate to use 0.0.

**DERFY** -  $D^3Y/DS^3$  of the airfoil upper surface at the trailing edge ( $XI=1.0$ ). It is usually sufficiently accurate to use 0.0.

**NIB** - The number of coordinate pairs used to describe the lower surface of the airfoil. Presently limited to 99.

**XIB(I)** - Input coordinates in the horizontal direction for the airfoil lower surface. The leading edge corresponds to  $XIB=0.0$  and the trailing edge is  $XIB=1.0$ .

**YIB(I)** - Input coordinates in the vertical direction for the airfoil lower surface. Since positive is up, the values of YIB are usually negative.

**DERIXB** -  $DX/DS$  of the airfoil lower surface at the leading edge. It is usually 0.0

**DERIYB** -  $DY/DS$  of the airfoil lower surface at the leading edge. It usually is -1.0

**DERFXB** -  $D^3X/DS^3$  of the airfoil lower surface at the trailing edge. It is usually sufficiently accurate to use 0.0.

**DERFYB** -  $D^3Y/DS^3$  of the airfoil lower surface at the trailing edge. It is usually sufficiently accurate to use 0.0.

**CPU(I)** - Upper surface inverse region  $C_p$  values for design case. II, which is computed internally, is the first grid point after XI. The distribution must be read in for each grid solved inversely (steps 14,17, and 20).

**CPL(I)** - Lower surface inverse  $C_p$  values for design case. They must be read in for each grid solved inversely (steps 15,18 and 21).

The Program Output for each Grid is:

- 1.) Heading
- 2.) Case Number
- 3.) Coordinate System. It is printed as I,X(I) followed by J,Y(J). Also, the Mach Number, angle of attack, and the location where the direct calculation stops is printed (i.e. X1).
- 4.) Listing of input data in namelists FINP and IINP.
- 5.) Airfoil ordinates in direct region
  - X - horizontal ordinate, where -0.5 is leading edge and 0.5 is trailing edge
  - YU - Upper surface ordinate
  - YL - Lower surface ordinate
- 6.) Desired  $C_p$  distribution in inverse region. Only printed in the inverse design case.
- 7.) Iteration history at ten-cycle intervals.
  - CIR - circulation
  - DPM - maximum  $\phi$  correction (absolute value) in the last relaxation cycle  
The (I,J) grid location of DPM is also printed.
  - NSSP - Number of supersonic points.
  - DELST -  $\delta^*$  at last grid point on upper surface. In interaction cases it is an indicator of convergence of solution.
  - DELTAY - Maximum absolute change in inverse region of the computational plane displacement surface ordinates during last surface update.
- 8.) Boundary layer details etc. - Every LP cycles details of the boundary layer calculation, current surface ordinates in computational coordinates, and surface slopes are printed. Useful if diagnostics needed.
- 9.) Final Boundary Layer Results (Viscous interaction case only)
  - YUORIG - Airfoil upper surface ordinate.
  - DU - Upper surface displacement thickness
  - SLU - Slope of upper displacement surface
  - YLORIG - Airfoil lower surface ordinate
  - SLL - Slope of lower displacement surface
  - DL - Lower surface displacement thickness

- 10.) Pressure Distribution on Airfoil
- 11.) Displacement Surface Ordinates and Slopes - In the inviscid case this will be the same as airfoil ordinates etc.
- 12.) Mach number chart of the flow field in the computational plane. Numbers printed are the Mach number multiplied by 100. I increases from top to bottom. J increases from left to right.
- 13.) Wave Drag Coefficient
- 14.) Plot of Results

U - Upper surface  $C_p$   
L - Lower surface  $C_p$   
T - Upper displacement surface  
B - Lower displacement surface  
CLCIR - Lift coefficient from circulation  
CL - Lift coefficient from integration of  $C_p$   
CD - CDWAVE + CDF  
CMLE - Moment coefficient about leading edge  
CDF - Skin friction drag coefficient  
CMC4 - Moment coefficient about quarter-chord

In addition, in the inverse case the following is printed after the final grid results.

- 15.) Boundary layer details - Upper surface
- 16.) Airfoil ordinates of Upper surface  
YOLD - Displacement surface ordinate  
YNEW - Airfoil ordinate  
DELSTAR - Displacement thickness
- 17.) Boundary layer details - Lower surface
- 18.) Airfoil ordinates of lower surface.

## APPENDIX A

### COORDINATE STRETCHING FUNCTIONS

To facilitate the application of the infinity boundary condition, the coordinates are stretched from a physical  $x$ - $y$  plane to the computational  $\xi$ - $\eta$  plane. To do this, the  $x$ -axis is subdivided into three regions. The first is from  $x = -\infty$  to  $x = -x_4$ . The second is from  $x = -x_4$  to  $x = x_4$ , and the last is from  $x = x_4$  to  $x = +\infty$ .

The stretching is symmetrical about the origin and is given by

$$x = x_4 + A_2 \tan \left[ \frac{\pi}{2}(\xi - \xi_4) \right] + A_3 \tan \left[ \frac{\pi}{2}(\xi - \xi_4)^3 \right]$$

in the third region and by

$$x = \xi(a + b\xi^2)$$

in the second region. The constants  $a$  and  $b$  are automatically computed by the program to satisfy the requirements

$$x = x_4 \text{ at } \xi = \xi_4$$

and

$$\frac{dx}{d\xi} = \frac{\pi A_2}{2} \text{ at } \xi = \xi_4$$

The constant  $A_2$  controls the grid spacing in the vicinity of  $x_4$ , near the leading and trailing edges of the airfoil, i.e.  $\Delta x \approx \frac{\pi A_2}{2} \Delta \xi$  at  $x=x_4$ .

$A_3$  determines the physical location of the grid line adjacent to the grid edge.

In the  $y$ -direction the stretching relationship is given by

$$y = A_1 \tan \left( \frac{\pi}{2} \eta \right)$$

and thus,  $A_1$  controls the grid size near the airfoil via

$$\frac{dy}{d\eta} = \frac{\pi A_1}{2} \sec^2 \left( \frac{\pi}{2} \eta \right)$$

and

$$\Delta y = \frac{\pi A_1}{2} \Delta \eta \text{ at } \eta = 0$$

Notice that these stretchings map the infinite  $x$ , $y$  plane into the finite computational plane

$$-(1 + \xi_4) \leq \xi \leq (1 + \xi_4)$$

$$-1 \leq \eta \leq 1$$

where  $\xi_4$  determines the amount of the computational plane confined to the vicinity of the airfoil. Also

$$\Delta\xi = \frac{2(1+\xi_4)}{IMAX-1}$$

and

$$\Delta\eta \approx \frac{2.0}{JMAX-1}$$

Finally, it should be noted that JMAX needs to be sufficiently large so that all points on the J=2 and J=JMAX-1 grid lines are subsonic. Otherwise, the rotated difference scheme may attempt to use points outside the computational space.

## APPENDIX B

### VISCOUS BOUNDARY LAYER AND WAVE DRAG CORRECTION

Experimental evidence indicates that viscous boundary layer effects are very important in transonic flow. For example, an aft-cambered airfoil inviscidly designed to have a lift coefficient of 0.6 may actually develop 25-50% less lift. This loss in lift is due not only to the existence of a boundary layer displacement surface but also to such factors as wake curvature and vertical pressure gradients in the trailing edge region. To prevent such discrepancies, the effect of the viscous boundary layer should be included in both the analysis and design portion of any numerical method.

In the present program, the approach is to assume that the inviscid streamlines follow a displacement surface having ordinates and slopes different from the actual airfoil. The effect of the fact that the streamlines do not follow a displacement surface in the vicinity of the trailing edge and that they are influenced by wake curvature is assumed to be either secondary or capable of being handled empirically. In the design case, the approach is to treat the airfoil determined by the inverse method as the displacement surface and to subtract from it the displacement thickness determined by a boundary layer computation. The result is considered to be the actual airfoil ordinates. For the analysis case, the approach is to calculate a boundary layer displacement surface (i.e. airfoil ordinate plus  $\delta^*$ ) using under-relaxation. The inviscid flowfield is then solved, and the displacement surface is updated every ten iterative cycles.

Obviously, the boundary layer scheme must be reliable, reasonably accurate, and computationally very efficient. After extensive investigation, the Nash-Macdonald method (ref.5) together with certain smoothing operations, was selected for incorporation into the present program. In addition, the displacement thickness at the trailing edge grid point was always determined by linear extrapolation from the previous two upstream grid point values. As a result, the basic approach used in the present program is similar to that of reference 6.

To update the displacement surface, the momentum integral equation

$$\frac{d\theta}{ds} + (H + 2M^2) \frac{\theta}{q} \frac{dq}{ds} = \tau$$

is solved for the momentum thickness  $\theta$  using the formulas of Nash and Macdonald

for skin friction,  $\tau$ , and the shape factor  $H = \delta^*/\theta$ . This computation is performed on the same grid spacing as the corresponding inviscid solution. The resultant displacement thickness is then smoothed everywhere and extrapolated to obtain the thickness at the trailing edge point. The smoothing is performed twice on grids having  $IMAX$  less than 55 and four times on grids with  $IMAX$  greater than 55.

This smoothing and extrapolation process appears to have two consequences. First, it reduces the rapid variations in the solution which sometimes occur in regions with high pressure gradients. Second, based on comparisons with experiments, the Nash-Macdonald method with smoothing and extrapolation seems to yield a trailing edge behavior that is correct with respect to the effect of the boundary layer on pressure distribution and lift. Admittedly, this behavior is fortuitous and some sensitivity to grid spacing has been detected. However, it should serve as a reasonable engineering model until a more complete, rational, trailing-edge theory is available. At that time such a theory could be easily incorporated into the present program.

If a case with extensive upper surface separation is encountered, the user may need to incorporate the optional trailing edge correction feature of the program. In this correction the boundary layer is solved using a modified pressure distribution that is linear from a point  $XLSEP$ , corresponding to separation, to the trailing edge. The base pressure, which determines the pressure gradient in this region, is determined semi-empirically. For aft-cambered airfoils, it is automatically selected to be the same as the maximum value of the pressure coefficient encountered on the lower surface of the airfoil. For conventional airfoils, it should be selected by the user based upon experience. (A typical value is 0.6). For aft-cambered airfoils the resultant modified boundary layer computation is normally applied only to the upper surface and is only used to determine the ordinate and slope of the displacement surface at  $XLSEP$ . The slope is then assumed to be constant from  $XLSEP$  to the trailing edge and the resultant displacement surface shape determined. Based upon comparisons with experiment, this approach yields reasonably good results and eliminates oscillations in the pressure distribution which can occur due to very small changes in the displacement surface slopes. Thus the method is a combination of the approaches used in references 6 and 7.

For conventional airfoils, the modified boundary layer computation is used to determine when the slope of the displacement surface becomes zero. From that point to the trailing edge the slope is then held constant. This approach is based on the concept that the streamlines from both the upper and lower surfaces should enter the wake almost parallel. Thus, for conventional airfoils, if the trailing edge correction is used, it should be applied to both surfaces. Since in most cases, this correction is not needed on conventional airfoils, its extensive use is not recommended until it has been verified by experiment.

In both the normal and corrected uses, separation is assumed to occur when  $(-\theta/q) dq/ds$  is greater than 0.004.

Now one of the difficulties associated with using a cartesian grid is that such a grid does not place a large number of computational points near the leading and trailing edges. Thus, the wave drag coefficient, which is determined by integration of the pressure distribution, has an inherent error associated with grid size, grid spacing, and the magnitude of the lift coefficient. Extensive comparisons with experimental data has indicated, however, that accurate estimates of the wave drag can be obtained by applying a suitable correction factor, CDCORR. This correction factor, which is different for each airfoil and computational grid, can be determined as a function lift from a series of calculations at different angles of attack at subcritical speeds, where the wave drag should be zero.

For each subcritical calculation, using a CDCORR of zero, determine the axial and normal coefficients using

$$CN = CL \cos \alpha + CDWAVE \sin \alpha$$

$$CA = -CL \sin \alpha + CDWAVE \cos \alpha$$

where CL is the lift coefficient determined from integration of the  $C_p$  distribution. Then the true CDCORR corresponding to CL can be computed from

$$CDCORR = CA + CN \sin \alpha / \cos \alpha$$

This value can then be used in supercritical runs having the same CL.

In some cases, it may be more convenient to compute the supercritical flows using a CDCORR of zero and to apply the correction later. In those cases, the following procedure can be used to determine the drag. First, compute the axial and normal coefficients using

$$CN = CL \cos \alpha + CDWAVE \sin \alpha$$

$$CA = -CL \sin \alpha + CDWAVE \cos \alpha$$

Then correct the axial coefficient for the appropriate lift and grid by

$$CA = CA - CDCORR$$

and recompute CDWAVE by

$$CDWAVE = CN \sin \alpha + CA \cos \alpha$$

The total drag coefficient is then given by

$$CD = CDWAVE + CDF$$

In all cases the drag due to skin friction and to changes resulting from the displacement surface shape is computed using the Squire-Young formula. While this formula is not exactly theoretically correct, it has been found to yield very accurate predictions.

APPENDIX C  
PROGRAM LISTING

```

C ***** TRANDES-- TRANSONIC ANALYSIS AND DESIGN PROGRAM *****
C ***** LELAND A. CARLSON, TEXAS A&M UNIVERSITY, 713-845-7541****00000002
C ***** JULY 1976*****00000003
C
C
      REAL M
      DIMENSION NTITLE(20),AA(500),IONIC(99)
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)00000007
      1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),
      1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(9900000009
      2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),
      3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,00000011
      4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP100000012
      5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JAI,JB1, 00000014
      1JMAX,JCON,JMAX1,NSSP,IW
      COMMON/JS/GG(99),GGP12(99),GGM12(99),GGM32(99),GGP32(99),A3 00000016
      COMMON/FIX/MHALF
      COMMON/ADelta/ITER
      COMMON/TAMU/DELTAY
      COMMON/REQ/ITERP
      COMMON/FIPUT/IREAD
      COMMON/NASH/RN,IBDLY,ITACT,YUORIG(99),YLORIG(99),SUPPER(99),SLOWER00000022
      1(99),DEL(99),DUPOLD(99),DLWOLD(99),CDF
      COMMON/IPT1/XIBDLY,RDEL,RDELFN,RCPB,SP,XSEP,CONV,CPB,XMON,XLSEP,
      1 MITER,LW,ITEUPC,ITELWC,XPC
      NAMELIST/FINP/M,W,X1,X2,ALP,EPS,EPSS,X4,S4,CONV,A1,A2,A3,RN,
      1 XIBDLY,CIR,CDCORR,RDEL,RDELFN,SP,XSEP,RCPB,CPB,XMON,XLSEP,XPC
      NAMELIST/IINP/IMAX,JMAX,IKASE,INV,MITER,NHALF,ITACT,ISKP2,ISKP3,
      1 ISKP4,ITERP,IREAD,LW,ITEUPC,ITELWC
      DELTAY=0.0
      CDF=0.
      CDCGRR=0.0
      DPM=0.0
      ITER=0
      MHALF=1
      IW=0
      CIR=0.0
      DPMSUM=0.0
      IDPM=1

```

DPOLD=0.0	00000040
M=0.5	00000041
W=1.7	00000042
X1=0.5	00000043
X2=100000.0	00000044
ALP=0.0	00000045
EPS=0.0	00000046
EPSS=0.4	00000047
X4=0.49	00000048
S4=2.0	00000049
A1=0.246	00000050
A2=0.15	00000051
<b>A3=3.87</b>	00000052
RN=20.0E+06	00000053
XIBDLY=-.44	00000054
CPB=0.4	00000055
RDEL=0.25	00000056
RDELFN=0.125	00000057
SP=0.004	00000058
XSEP=0.44	00000059
XLSEP=0.50	00000060
RCPB=0.2	00000061
XMON=0.47	00000062
CONV=1.E-05	00000063
I MAX=13	00000064
J MAX=7	00000065
IKASE=100	00000066
INV=0	00000067
MITER=800	00000068
NHALF=2	00000069
ITACT=0	00000070
ISKP2=0	00000071
ISKP3=0	00000072
ISKP4=0	00000073
ITERP=0	00000074
IREAD=0	00000075
LP=1000	00000076
ITEUPC=0	00000077
ITELWC=0	00000078
XPC=0.10	00000079

```

      READ(5,1)(NTITLE(I),I=1,20)          00000080
1 FORMAT(20A4)                         00000081
      PRINT 2                           00000082
2 FORMAT(1H1)                           00000083
      READ(5,FINP)
      EPSSO=EPSS
      EPSO=EPS
      BETA=SQRT(1.-M*M)
      READ(5,IINP)
      ICASE=IKASE
8001 CONTINUE                           00000086
      ALPDEG=ALP
      CPSTAR=1.428/(M*M)*(((1.+0.2*M*M)/1.2)**3.5-1.)
      PI=4.*ATAN(1.0)                   00000087
      PI2=0.5*PI                        00000088
      ALP=ALP*PI/180.
      A22=2./(PI*A2)                   00000089
      A11=2./(PI*A1)
      CALL VALUE                         00000090
101 PRINT 3,(NTITLE(I),I=1,20)          00000091
3 FORMAT(20A4)                         00000092
      CALL COORD
      PRINT 6,M,ALPDEG,X1,ICASE        00000093
6 FORMAT(1H0,3X,'MACH NO. IS ',F5.3,' ANGLE OF ATTACK IS ',F5.3,' DE00000103
      1GREES',//,10X,' DIRECT SOLUTION TO ',F8.2,           // 00000104
      225X,'CASE NUMBER',I6)           00000105
      IF(INV.EQ.0)PRINT 6001           00000106
6001 FORMAT(1H0,3X,'INVISCID ANALYSIS CASE')
      IF(ITACT.EQ.1)PRINT 6002           00000107
6002 FORMAT(1H ,3X,'WITH VISCOUS INTERACTION')
      IF(INV.EQ.1)PRINT 6003           00000108
6003 FORMAT(1H0,3X,'INVERSE DESIGN CASE')
      WRITE(6,FINP)                     00000109
      WRITE(6,IINP)                     00000110
      IF(MHALF.EQ.1)GO TO 102          00000111
      JB=JMAX/2+1                      00000112
      DO 104 I=ILE,IMAX                00000113
                                         00000114
                                         00000115
                                         00000116

```

```

104 P(I,JB-1)=0.5*(P(I,JB-2)+PB(I))          00000117
      P(IMAX,JMAX-1)=P(IMAX,JMAX-2)           00000118
      P(IMAX-1,JMAX)=P(IMAX-2,JMAX)           00000119
      P(2,JMAX)=P(3,JMAX)                      00000120
      P(1,JMAX-1)=P(1,JMAX-2)                  00000121
      P(1,2)=P(1,3)                           00000122
      P(2,1)=P(3,1)                           00000123
      P(IMAX-1,1)=P(IMAX-2,1)                  00000124
      P(IMAX,2)=P(IMAX,3)                      00000125
102 CCNTINUE                                     00000126
      ILE1=ILE-1                                00000127
      I11=I1-1                                  00000128
      ITE=IMAX-ILE1                            00000129
      ITE1=ITE+1                               00000130
      CALL FCIL                                 00000131
      IF(IREAD.EQ.1 .AND. MHALF.EQ.1)MHALF=MHALF+1 00000132
7 DO 8 J=1,JMAX                                 00000133
8 P1(J)=P(1,J)                                00000134
      CALL FLOW1                                00000135
      CALL FLOW2                                00000136
      IF(INV.EQ.0)GO TO 9                      00000137
      IF(MHALF.EQ.1)GO TO 9                      00000138
      CALL FLOW3                                00000139
      IF(X2.GT.1000.0)GO TO 10                   00000140
9 CALL WAKE                                    00000141
10 STE=S4+2./PI*ATAN((0.5-X4)/A2)            00000142
      CIR=-(STE-S(ITE+1))/DS*(P(ITE,JB)-PB(ITE))+(STE-S(ITE))/DS*
1(P(ITE+1,JB)-PB(ITE+1))                     00000143
      QUAN1=-.5*CIR/PI                         00000144
      QUAN2=ATAN(BETA*A1/A2*DS/DE)             00000145
      QUAN3=ATAN(BETA*TAN(ALP))                00000146
      IF(M.GT.1.)GO TO 11                      00000147
      IF(ALP.GT.0.0)GO TO 108                  00000148
      IF(ALP.LT.0.0)GO TO 1081                 00000149
      P(IMAX,JMAX)=QUAN1*QUAN2               00000150
      DO 12 I=2,IMAX1                          00000151
      P(I,JMAX)=- CIR/4.0                      00000152
12 P(I,1)=-0.75*CIR                         00000153
      P(1,JMAX)=QUAN1*(PI-QUAN2)              00000154
                                         00000155

```

```

P(1,1)=QUAN1*(PI+QUAN2)          00000156
P(IMAX,1)=QUAN1*(2.*PI-QUAN2)    00000157
DO 13 J=2,JMAX1                  00000158
13 P(1,J)=-0.5*CIR              00000159
JBM1=JB-1                         00000160
DO 14 J=2,JBM1                  00000161
14 P(IMAX,J)=-CIR              00000162
GO TO 109                         00000163
1081 DO 1083 I=2,IMAX1           00000164
P(I,JMAX)=QUAN1*(PI+ATAN(BETA/TAN(ALP))) 00000165
1083 P(I,1)=QUAN1*PI+P(I,JMAX)   00000166
GO TO 1082                         00000167
108 DO 110 I=2,IMAX1           00000168
P(I,JMAX)=QUAN1*ATAN(BETA/TAN(ALP)) 00000169
110 P(I,1)=QUAN1*PI+P(I,JMAX)   00000170
1082 CONTINUE                      00000171
DO 111 J=2,JMAX1                  00000172
111 P(1,J)=QUAN1*(PI-QUAN3)     00000173
JBM1=JB-1                         00000174
DO 112 J=2,JBM1                  00000175
112 P(IMAX,J)=QUAN1*(2.*PI-QUAN3) 00000176
DO 113 J=JB,JMAX1                  00000177
113 P(IMAX,J)=QUAN1*(-QUAN3)     00000178
P(IMAX,JMAX)=QUAN1*QUAN2         00000179
P(1,JMAX)=QUAN1*(PI-QUAN2)       00000180
P(1,1)=QUAN1*(PI+QUAN2)         00000181
P(IMAX,1)=QUAN1*(2.*PI-QUAN2)    00000182
109 CONTINUE                      00000183
PB(IMAX)=-CIR+P(IMAX,JB)        00000184
11 ITER=ITER+1                    00000185
DPMSUM=DPMSUM+DPM               00000186
IDPM=IDPM+1                      00000187
IF(IDPM.LE.10)GO TO 512          00000188
DPOLD=DPMSUM                      00000189
DPMSUM=0.0                         00000190
IDPM=1                           00000191
512 CONTINUE                      00000192
IF(ITACT.EQ.1)DELTAY=DUPOLD(ITE) 00000193
IF(ITER/10*10.EQ.ITER)           00000194

```

```

1PRINT 15,ITER,CIR,DPM,ICON,JCON,NSSP,DELTAY          00000195
15 FORMAT(1H , 'ITERATION',I4,' CIR = ',F8.5,' DPM = ',F11.8,' AT',
12I3,' NSSP = ',I4,' DELTAY OR DELSTAR = ',F7.4)      00000196
12I3,' NSSP = ',I4,' DELTAY OR DELSTAR = ',F7.4)      00000197
12I3,' NSSP = ',I4,' DELTAY OR DELSTAR = ',F7.4)      00000198
IF(M.LE.1.0)GO TO 16
C ADD P(IMAX,J) CARD HERE FOR M GT 1.0 CASE          00000199
16 IF(INV.EQ.0.AND.ITACT.EQ.0)GO TO 24               00000200
IF(MHALF.EQ.1)GO TO 24                               00000201
IF(IREAD.EQ.1.AND.ITACT.EQ.1)GO TO 9005            00000202
IF(ITER.LT.50)GO TO 24                               00000203
IF(ITACT.EQ.1)GO TO 9005                           00000204
IF(MHALF.EQ.2.AND.ISKP2.EQ.1)GO TO 24             00000205
IF(MHALF.EQ.3.AND.ISKP3.EQ.1)GO TO 24             00000206
IF(MHALF.EQ.4.AND.ISKP4.EQ.1)GO TO 24             00000207
IF(ITER/10*10.EQ.ITER)CALL SHAPE                  00000208
GO TO 9006                                         00000209
9005 IF(ITER/10*10.EQ.ITER)CALL VISACT            00000210
9006 IF(ITER/LP*LP.EQ.ITER)PRINT22,(X(I),YU(I),YL(I),I=ILE,IMAX1)
IF(ITER/LP*LP.EQ.ITER)PRINT 22,(X(I),SLU(I),SLL(I),I=ILE,IMAX1) 00000211
24 CONTINUE                                         00000212
IF(ITER.GE.MITER)GO TO 17                         00000213
IF(INV.EQ.0.AND.ITACT.EQ.0)GO TO 106            00000214
IF(MHALF.GT.1)GO TO 106                         00000215
IF(MHALF.EQ.1.AND.ITER.LT.50)GO TO 106           00000216
DPM=0.                                           00000217
00000218
106 CONTINUE                                         00000219
IF(DPM.LT.CONV)GO TO 17                         00000220
21 DPM=0.0                                         00000221
GO TO 7                                         00000222
17 CONTINUE                                         00000223
C *** THE FOLLOWING CAN BE USED TO PRINT OUT THE *****
C *****PERTURBATION POTENTIAL FLOWFIELD SOLUTION IF DESIRED*****
C DO 18 JJ=1,JMAX                                00000224
C JJ=JMAX+1-JJ                                     00000225
C PRINT 19,J                                       00000226
C PRINT 19,J                                       00000227
C PRINT 19,J                                       00000228
C 19 FORMAT(1H , 'ROW  ',I5)                      00000229
C PRINT 20,(P(I,J),I=1,IMAX)                     00000230
C 20 FORMAT(1H ,10E11.3)                          00000231
C 18 CONTINUE                                         00000232

```

```

C      PRINT 19,JB          00000233
C      PRINT 20,(PB(I),I=1,IMAX) 00000234
C      IF(MHALF.LE.NHALF)GO TO 8007 00000235
C      ***** THE FOLLOWING CAN BE USED TO PUNCH OUTPUT IF DESIRED*****
C      DO 8005 JJ=1,JMAX          00000236
C      C8006 FORMAT(5E15.7)        00000237
C      J=JMAX-JJ+1               00000238
C      C8005 CCONTINUE          00000239
C      8007 CCONTINUE          00000240
C      PRINTE SHAPE HERE IF REQUIRED 00000241
C      IF(MHALF.EQ.1)GO TO 26      00000242
C      IF(INV.EQ.1)CALL SHAPE     00000243
C      IF(ITACT.EQ.0)GO TO 26      00000244
C      IF(ITER.GE.MITER)GO TO 7501 00000245
C      CALL VISACT              00000246
7501 PRINT 9008,RN          00000247
9008 FORMAT(*0*, "BOUNDARY LAYER ANALYSIS FOR REYNOLDS NUMBER OF", E12.3, 00000248
    *//,5X,"X",9X,"YUORIG",4X,"DU",8X,"SLU",7X,"YLORIG",4X,"DL",8X,"SLL",00000250
    *)                           00000251
    PRINT 9009,(X(I),YUORIG(I),DUPOLD(I),SLU(I),YLORIG(I),DLWOLD(I), 00000252
    *SLL(I),I=ILE,ITE)           00000253
9009 FORMAT(5X,7F10.5)        00000254
    GO TO 9007                  00000255
    26 CALL PRESS                00000256
9007 DO 25 I=ILE,IMAX1       00000257
    YU(I)=A1*TAN(PI/2.*YU(I))   00000258
    25 YL(I)=A1*TAN(PI/2.*YL(I)) 00000259
    PRINT 6004                  00000260
6004 FORMAT(1H , " CP BY CENTRAL DIFFERENCES") 00000261
    PRINT 9010                  00000262
9010 FORMAT(1H , "X",10X,"CPU",10X,"CPL") 00000263
    PRINT 9011,(X(I),CPU(I),CPL(I),I=ILE,IMAX1) 00000264
9011 FORMAT(1H , 3F10.3)        00000265
    IMAX2=IMAX-2               00000266
    PRINT 221                  00000267
221 FORMAT(1H , "X",10X,"YU",10X,"YL",10X,"SLU",8X,"SLL") 00000268
    22 FORMAT(3(' X= ',F7.4,' YU= ',F7.4,' YL= ',F7.4)) 00000269
    PRINT 220,(X(I),YU(I),YL(I),SLU(I),SLL(I),I=ILE,ITE) 00000270
220 FORMAT(1H , 5F10.5)        00000271
    IF(MHALF.LE.NHALF)GO TO 8014 00000272

```

```

8014 CONTINUE
DO 9002 I=ILE,ITE
YU(I)=ATAN(YU(I)/A1)/PI2
9002 YL(I)=ATAN(YL(I)/A1)/PI2
DO 500 I=2,ILE1
DO 501 J=2,JMAX1
U=QI*(COS(ALP)+FF(I)*(P(I+1,J)-P(I-1,J))/(2.*DS))
V=QI*(SIN(ALP)+GG(J)*(P(I,J+1)-P(I,J-1))/(2.*DE))
UU=U*U
VV=V*V
AD=A12-0.2*(UU+VV-QI2)
501 IONIC(J)=100.0*SQRT((UU+VV)/AD)
PRINT 28,(IONIC(J),J=2,JMAX1)
500 CONTINUE
DO 502 I=ILE,ITE
DO 503 J=2,JMAX1
503 ICNIC(J)=0
JB2=JB-2
DO 504 J=JB2,JMAX1
IF(YU(I).GT.E(J).AND.YU(I).LE.E(J+1))GO TO 505
504 CONTINUE
505 JA=J+1
IF(JA.LE.JB)JA=JB+1
DO 506 J=JA,JMAX1
U=QI*(COS(ALP)+FF(I)*(P(I+1,J)-P(I-1,J))/(2.*DS))
V=QI*(SIN(ALP)+GG(J)*(P(I,J+1)-P(I,J-1))/(2.*DE))
UU=U*U
VV=V*V
AD=A12-0.2*(UU+VV-QI2)
506 IONIC(J)=100.0*SQRT((UU+VV)/AD)
JB2=JB+2
DO 507 JJ=1,JMAX1
J=JB2-JJ
IF(YL(I).GE.E(J).AND.YL(I).LT.E(J+1))GO TO 508
507 CONTINUE
508 JA=J
IF(JA.GE.JB)JA=JB-1
DO 509 J=2,JA
U=QI*(COS(ALP)+FF(I)*(P(I+1,J)-P(I-1,J))/(2.*DS))
V=QI*(SIN(ALP)+GG(J)*(P(I,J+1)-P(I,J-1))/(2.*DE))

```

00000273  
00000274  
00000275  
00000276  
00000277  
00000278  
00000279  
00000280  
00000281  
00000282  
00000283  
00000284  
00000285  
00000286  
00000287  
00000288  
00000289  
00000290  
00000291  
00000292  
00000293  
00000294  
00000295  
00000296  
00000297  
00000298  
00000299  
00000300  
00000301  
00000302  
00000303  
00000304  
00000305  
00000306  
00000307  
00000308  
00000309  
00000310  
00000311  
00000312

```

IF(J.EQ.(JB-1))V=QI*(SIN(ALP)+GG(J)*(PB(I)-P(I,J-1))/(2.*DE))      00000313
UU=U*U
VV=V*V
AD=A12-0.2*(UU+VV-QI2)
509 IONIC(J)=100.0*SQRT((UU+VV)/AD)                                     00000317
PRINT 28,(IONIC(J),J=2,JMAX1)                                              00000318
502 CONTINUE                                                               00000319
DO 510 I=ITE1,IMAX1                                                       00000320
DO 511 J=2,JMAX1                                                       00000321
U=QI*(COS(ALP)+FF(I)*(P(I+1,J)-P(I-1,J))/(2.*DS))                  00000322
V=QI*(SIN(ALP)+GG(J)*(P(I,J+1)-P(I,J-1))/(2.*DE))                  00000323
IF(J.EQ.JB)V=V-QI*(GG(J)*(CIR/(2.*DE)))
IF(J.EQ.JB-1)V=V-QI*(GG(J)*(CIR/(2.*DE)))
UU=U*U
VV=V*V
AD=A12-0.2*(UU+VV-QI2)
511 IONIC(J)=100.0*SQRT((UU+VV)/AD)                                     00000329
PRINT 28,(IONIC(J),J=2,JMAX1)                                              00000330
510 CONTINUE                                                               00000331
28 FORMAT(1H ,40I3)                                                       00000332
DO 9003 I=ILE,ITE                                                       00000333
YU(I)=A1*TAN(PI/2.*YU(I))                                                 00000334
9003 YL(I)=A1*TAN(PI/2.*YL(I))                                             00000335
CL=0.5*(CPL(ILE)-CPU(ILE))*(X(ILE)+0.5)                                 00000336
CPSTAG=2.0/(1.4*M*M)*((1.0+0.2*M*M)**3.5-1.0)                         00000337
CD=(CPSTAG+CPU(ILE))*YU(ILE)*0.5-(CPSTAG+CPL(ILE))*(YL(ILE))*0.500000338
CMLE=0.5*(CPU(ILE)-CPL(ILE))*(X(ILE)+0.5)**2                           00000339
IEND=ITE-1                                                               00000340
DO 9000 I=ILE,IEND                                                       00000341
T1=CPL(I)-CPU(I)                                                       00000342
T2=CPL(I+1)-CPU(I+1)                                                 00000343
T3=(X(I+1)-X(I))*0.5                                                 00000344
CL=CL+(T1+T2)*T3                                                 00000345
IF(ITACT.EQ.1)GO TO 8010                                              00000346
CD=CD+(CPU(I)+CPU(I+1))*0.5*(YU(I+1)-YU(I))-(CPL(I)+CPL(I+1))*0.5 00000347
1*(YL(I+1)-YL(I))                                                 00000348

```

```

      GO TO 8011                               00000349
8010 CD=CD+((CPU(I)+CPU(I+1))* .5*(YUORIG(I+1)-YUORIG(I))-(CPL(I)+CPL(I+1)*00000350
     *))* .5*(YLORIG(I+1)-YLORIG(I))          00000351
8011 CONTINUE                                00000352
      T6=-T1*(X(I)+0.5)                      00000353
      T7=-T2*(X(I+1)+0.5)                      00000354
9000 CMLE=CMLE+(T6+T7)*T3                   00000355
      CL=CL+0.5*(CPL(ITE)-CPU(ITE))*(0.5-X(ITE)) 00000356
      CMLE=CMLE+0.5*(CPU(ITE)-CPL(ITE))*(X(ITE)+0.5)*(0.5-X(ITE)) 00000357
      IF(ITACT.EQ.1)GO TO 8012                00000358
      CD=CD-((CPU(ITE)+CPL(ITE))*0.5*(YU(ITE)-YL(ITE))) 00000359
      GO TO 8013                                00000360
8012 CD=CD-@DCORR                           00000361
8013 CONTINUE                                00000362
      FN=CL*COS(ALP)-CD*SIN(ALP)            00000363
      FT=CL*SIN(ALP)+CD*COS(ALP)            00000364
      CL=FN                                    00000365
      CD=FT                                    00000366
      CMC4=CMLE+CL/4.                         00000367
      CDWAV=CD                                 00000368
      CD=CDWAV+CDF                            00000369
      PRINT 9012,CDWAV                          00000370
9012 FORMAT(1H0.20X,*WAVE CD = ',F10.6)       00000371
      NOV=ITE-ILE+1                           00000372
      DO 114 I=ILE,ITE                         00000373
      J=I-ILE1                                00000374
      J1=J+NOV                                00000375
      J2=J+2*NOV                             00000376
      J3=J+3*NOV                             00000377
      J4=J+4*NOV                             00000378
      AA(J)=X(I)                             00000379
      AA(J1)=CPU(I)                           00000380
      AA(J2)=CPL(I)                           00000381
      AA(J3)=-YU(I)                           00000382
114  AA(J4)=-YL(I)                           00000383
      NL=50                                  00000384
      IF(NOV.GT.45)NL=100                      00000385
      CALL PLOT(ICASE,AA,NOV,5,NL,0)           00000386
      CLCIR=2.*CIR                            00000387
      PRINT 8002,CPSTAR,CLCIR                 00000388

```

```

8002 FORMAT(1H ,40X,'PRESSURE COEFFICIENT',//,41X,'CPSTAR = ',F10.4,    00000389
  1 5X,'CLCIR = ',F10.4)                                         00000390
  PRINT 9001,CL,CD,CMLE,CDF,CMC4                                00000391
9001 FORMAT(1H0,20X,'CL = ',F10.4,' CD = ',F10.6,' CMLE = ',F10.4, 00000392
  ** CDF = ',F10.6,' CMC4 = ',F10.4)                           00000393
  PRINT 8003                                              00000394
8003 FORMAT(1H1)                                              00000395
  IF(MHALF.GT.NHALF)GO TO 100                                 00000396
  MHALF=MHALF+1                                              00000397
  MITER=MITER/2                                              00000398
  CALL HALVE                                              00000399
C                                                               00000400
  IF(INV.EQ.1.AND.MHALF.EQ.3)MITER=400
  IF(MHALF.EQ.4)MITER=400                                     00000401
C                                                               00000402
  DPM=0.0                                              00000403
  DPOLD=0.0                                              00000404
  IDPM=1                                               00000405
  DPMSUM=0.0                                              00000406
  EPSS=EPSS0                                             00000407
  EPS=EPS0                                              00000408
  ITER=0                                                 00000409
  DELTAY=0.0                                              00000410
  GO TO 101                                              00000411
100 CONTINUE                                              00000412
  IF(INV.EQ.1)CALL BDLY                                     00000413
  IF(ITACT.EQ.1)GO TO 9014                                 00000414
  WRITE(7,9015)(X(I),YU(I),YL(I),CPU(I),CPL(I),I=ILE,ITE) 00000415
9015 FORMAT(5F10.5)                                         00000416
  STOP                                              00000417
9014 WRITE(7,9015)(X(I),YUORIG(I),YLORIG(I),CPU(I),CPL(I),I=ILE,ITE) 00000418
  STOP                                              00000419
  END                                              00000420

```

```

SUBROUTINE FOIL                                00000421
CC **** READS IN INITIAL AIRFOIL SHAPE AND DETERMINES ORDINATES 00000422
C ***** AND SLOPES AT COMPUTATIONAL GRID POINTS ***** 00000423
      REAL M                                     00000424
      DIMENSION XI(99),YI(99),X0(99),Y0(99),SI(99),SO(99),XP(99),YP(99),00000425
      1DIY(99),D2Y(99),D3Y(99),XIB(99),YIB(99)          00000426
      DIMENSION XOR(99)                            00000427
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)00000428
      1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),00000429
      1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(9900000430
      2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),00000431
      3A1,A2,A12,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,00000432
      4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP100000433
      5,QQ,UUJB1,PI,PI2,A22,A11,X4,S4                00000434
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,00000435
      1JMAX,JCON,JMAX1,NSSP,IW                      00000436
      CCOMMON/FIX/MHALF                           00000437
      CCOMMON/RED/ITERP                          00000438
      CCOMMON/FINPUT/IREAD                     00000439
      COMMON/NASH/RN,IBDLY,ITACT,YUORIG(99),YLORIG(99),SUPPER(99),SLOWER00000440
      1(99),DEL(99),DUPOLD(99),DLWOLD(99),CDF          00000441
      CCOMMON/IPT1/XIBDLY,RDEL,RDELFN,RCPB,SP,XSEP,CONV,CPB,XMON,XLSEP,00000442
      1 MITER,LP,ITEUPC,ITELWC,XPC                 00000443
C THIS PROGRAM DEPENDS UPON AIRFOIL BEING BEING STUDIED 00000444
      PRINT 2                                      00000445
2 FORMAT(1H0,20X,'AIRFOIL COORDINATES',/,5X,'X           YU           YL 00000446
      1   UPPER SLOPE LOWER SLOPE')                  00000447
      IBDLY=ILE-1                                  00000448
215 IBDLY=IBDLY+1                            00000449
      IF(X(IBDLY).LT.XIBDLY)GO TO 215            00000450
      IF(ITACT.EQ.1)GO TO 35                      00000451
      IF(INV.EQ.0)GO TO 7                         00000452
35 IF(MHALF.LE.2)GO TO 7                      00000453
      I=IMAX1/2                                    00000454
      I1=IMAX-2                                    00000455
      ISTOR=I11                                    00000456
      IF(ITACT.EQ.1)I11=IBDLY-1                  00000457
8 CPU(I1)=CPU(I)                            00000458
      CPL(I1)=CPL(I)                            00000459
      SLU(I1)=SLU(I)                            00000460

```

```

SLL(II)=SLL(I)                                00000461
DUPOLD(II)=DUPOLD(I)                            00000462
DLWOLD(II)=DLWOLD(I)                            00000463
YU(II)=YU(I)                                    00000464
YL(II)=YL(I)                                    00000465
I=I-1                                         00000466
II=II-2                                       00000467
IF(II.GE.I11)GO TO 8                           00000468
IMAX2=IMAX-2                                  00000469
IS=II+3                                       00000470
DO 9 I=IS,IMAX2,2                             00000471
DUPOLD(I)=.5*(DUPOLD(I+1)+DUPOLD(I-1))       00000472
DLWOLD(I)=.5*(DLWOLD(I+1)+DLWOLD(I-1))       00000473
CPU(I)=0.5*(CPU(I+1)+CPU(I-1))                00000474
CPL(I)=0.5*(CPL(I+1)+CPL(I-1))                00000475
SLU(I)=0.5*(SLU(I+1)+SLU(I-1))                00000476
SLL(I)=0.5*(SLL(I+1)+SLL(I-1))                00000477
YU(I)=0.5*(YU(I+1)+YU(I-1))                  00000478
9 YL(I)=0.5*(YL(I+1)+YL(I-1))                00000479
YU(IMAX1)=0.0001                               00000480
YL(IMAX1)=-0.0001                             00000481
SLU(IMAX1)=0.0                                 00000482
SLL(IMAX1)=0.0                                 00000483
DUPOLD(IMAX1)=0.                                00000484
DLWOLD(IMAX1)=0.                                00000485
I11=ISTOR                                     00000486
GO TO 10                                      00000487
7 CONTINUE                                     00000488
DO 6 I=ITE1,IMAX                               00000489
YUORIG(I)=.0001                                00000490
YLORIG(I)=-0.0001                             00000491
DUPOLD(I)=0.                                    00000492
DLWOLD(I)=0.                                    00000493
YU(I)=0.0001                                   00000494
YL(I)=-0.0001                                  00000495
SLU(I)=0.0                                     00000496
6 SLL(I)=0.0                                   00000497
10 IF(INV.EQ.1)IEND=I11                         00000498
IF(INV.EQ.0)IEND=ITE                          00000499
IF(MHALF.LT.3)IEND=ITE                        00000500

```

```

C      UPPER SURFACE, XI IN PERCENT CHORD          00000501
      IF(MHALF.GT.1)GO TO 21                      00000502
      READ 14, NI                                    00000503
      READ 15,(XI(I),YI(I),I=1,NI)                 00000504
      READ15,DERIX,DERIY,DERFX,DERFY               00000505
      14 FORMAT(15)                                00000506
      DO 18 I=1,NI                                 00000507
      18 XI(I)=XI(I)-0.5                         00000508
      21 DO 16 I=ILE,IEND                         00000509
      II=I-ILE+1                                  00000510
      15 FORMAT(8F10.4)                            00000511
C   15 FORMAT(2F10.4)                            00000512
      16 X0(II)=X(I)                             00000513
      NO=IEND-ILE+1                            00000514
      CALL ARC(XI,YI,X0,Y0,SI,SD,XP,YP,D1Y,D2Y,D3Y,DERIX,DERFX,DERFY,
      1DERFY,NI,NO,1)                           00000515
      IF(ITACT.EQ.1)GO TO 23                     00000516
      DO 17 I=ILE,IEND                         00000517
      II=I-ILE+1                                  00000518
      DUPOLD(I)=0.                               00000519
      YU(I)=Y0(II)                                00000520
      YUORIG(I)=Y0(II)                           00000521
      17 SLU(I)=YP(II)/XP(II)                   00000522
      GO TO 25                                   00000523
      23 DO 24 I=ILE,IEND                         00000524
      II=I-ILE+1                                  00000525
      IF(I.LT.IBDLY)GO TO 36                     00000526
      IF(MHALF.GT.2)GO TO 26                     00000527
      36 YU(I)=Y0(II)                            00000528
      YUORIG(I)=Y0(II)                           00000529
      DUPOLD(I)=0.                               00000530
      SUPPER(I)=SO(II)                           00000531
      SLU(I)=YP(II)/XP(II)                   00000532
      GO TO 24                                   00000533
      26 YUORIG(I)=Y0(II)                           00000534
      SUPPER(I)=SO(II)                           00000535
      24 CONTINUE                                00000536
C      LOWER SURFACE, XI IN PERCENT CHORD        00000537
      25 IF(MHALF.GT.1)GO TO 22                  00000538
      READ14,NIB                                00000539

```

```

READ15,(XIB(I),YIB(I),I=1,NIB)          00000540
READ15,DERIXB,DERIYB,DERFXB,DERFYB      00000541
DO 19 I=1,NIB                           00000542
19 XIB(I)=XIB(I)-0.5                   00000543
22 CALL ARC(XIB,YIB,XO,YO,SI,SO,XP,YP,D1Y,D2Y,D3Y,DERIXB,DERFXB,
    1DERIYB,DERFYB,NIB,NO,1),             00000544
    IF(ITACT.EQ.1)GO TO 27               00000545
    DO 20 I=ILE,IEND                     00000546
    II=I-ILE+1                          00000547
    DLWOLD(I)=0.                         00000548
    YL(I)=YO(II)                        00000549
    YLORIG(I)=YL(II)                    00000550
20 SLL(I)=YP(II)/XP(II)                 00000551
    GO TO 28                           00000552
27 DO 29 I=ILE,IEND                     00000553
    II=I-ILE+1                          00000554
    IF(I.LT.IBDLY)GO TO 37              00000555
    IF(MHALF.GT.2)GO TO 30              00000556
37 YL(I)=YO(II)                        00000557
    YLORIG(I)=YO(II)                    00000558
    DLWOLD(I)=0.                         00000559
    SLOWER(I)=SO(II)                    00000560
    SLL(I)=YP(II)/XP(II)                00000561
    GO TO 29                           00000562
30 YLORIG(I)=YO(II)                    00000563
    SLOWER(I)=SO(II)                    00000564
29 CONTINUE                            00000565
    IF(IREAD.EQ.0)GO TO 28              00000566
    READ 217,(YU(I),YL(I),SLU(I),SLL(I),I=ILE,ITE) 00000567
    READ 217,(DUPOLD(I),DLWOLD(I),I=ILE,ITE)        00000568
217 FORMAT(5E15.7)                      00000569
28 DO 1 I=ILE,IEND                     00000570
    PRINT 3, X(I),YU(I),YL(I),SLU(I),SLL(I)       00000571
3 FORMAT(5F10.5)                       00000572
1 CONTINUE                             00000573
C   FINDING COORDS IN ETA-PSI SYSTEM   00000574
    DO 4 I=ILE,IMAX1                     00000575
    YU(I)=ATAN(YU(I)/A1)/PI2            00000576
4 YL(I)=ATAN(YL(I)/A1)/PI2            00000577
    IF(INV.EQ.0)RETURN                  00000578

```

IF(IREAD.EQ.1.AND.MHALF.EQ.1)GO TO 103	00000579
IF(MHALF.EQ.1)RETURN	00000580
IF(MHALF.GT.3.AND.ITERP.EQ.1)GO TO 100	00000581
103 CONTINUE	00000582
READ(5,5)(CPU(I),I=I1 ,ITE)	00000583
CPU(I11)=0.0	00000584
READ(5,5)(CPL(I),I=I1 ,ITE)	00000585
CPL(I11)=0.0	00000586
ITEP1=ITE+1	00000587
DO 339 I=ITEP1,IMAX	00000588
CPU(I)=0.	00000589
339 CPL(I)=0.	00000590
5 FORMAT(8F10.3)	00000591
100 CONTINUE	00000592
PRINT 101	00000593
101 FORMAT(1H0,20X,"UPPER CP INPUT")	00000594
PRINT 5,(CPU(I),I=I1 ,ITE)	00000595
PRINT 102	00000596
102 FORMAT(1H0,20X,"LOWER CP INPUT")	00000597
PRINT 5,(CPL(I),I=I1 ,ITE)	00000598
RETURN	00000599
END	00000600

```

SUBROUTINE VISACT                               00000601
C                                              00000602
C **** COMPUTES BOUNDARY LAYER WHEN VISCOUS INTERACTION INCLUDED 00000603
C***** IN THE ANALYSIS CASE *****               00000604
C                                              00000605
C
      REAL M,NEW                                00000606
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99) 00000607
      1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),                           00000608
      1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99)00000609
      2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),                           00000610
      3A1,A2,A12,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,00000611
      4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP100000612
      5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4                                00000613
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1, 00000614
      1JMAX,JCON,JMAX1,NSSP,IW                                         00000615
      COMMON/NASH/RN,IBDLY,ITACT,YUORIG(99),YLORIG(99),SUPPER(99),SLOWER00000616
      1(99),DEL(99),DUPOLD(99),DLWOLD(99),CDF                            00000617
      COMMON/DELTA/ITER                                         00000618
      COMMON/PIPT1/XIBDLY,RDEL,RDELFN,RCPB,SP,XSEP,CONV,CPB,XMON,XLSEP, 00000619
      1 MITER,LP,ITEUPC,ITEWC,XPC                                     00000620
      DIMENSION UE(99),DSS(99),DUDS(99),YUN(99),YLN(99),EM(99)          00000621
      DIMENSION XI(99),YI(99),X0(99),Y0(99),SI(99),SD(99),XP(99),YP(99),00000622
      1D1Y(99),D2Y(99),D3Y(99),XIB(99),YIB(99)                         00000623
      DIMENSION HS(99),XQR(99)                                         00000624
      DIMENSION CPUT(99),CPLT(99)                                       00000625
      SEPMK=0                                         00000626
      IF(IMAX.GT.55)RDEL=RDELFN                                     00000627
      ISIDE=0                                         00000628
      ICYCLE=1                                         00000629
      ICYBOT=1                                         00000630
      CALL PRESS                                         00000631
      IF(DPM.LE.CONV)GO TO 5009                                 00000632
      IF(ITER.GE.(MITER-1))GO TO 5009                           00000633
      GO TO 5005                                         00000634
5009 DO 5006 J=ILE,IMAX1                               00000635
      CPUT(J)=CPU(J)                                         00000636
5006 CPLT(J)=CPL(J)                                   00000637
5005 CONTINUE                                         00000638
      DO 500 J=ILE,ITE                                     00000639
      YU(J)=A1*TAN(PI/2.*YU(J))                           00000640

```

```

500 YL(J)=A1*TAN(PI/2.*YL(J))          00000641
TR=0.3424                                00000642
TE1=5.E-03                                00000643
TE2=5.E-05                                00000644
LMON=IMAX1/2+1                            00000645
4001 LMON=LMON+1                          00000646
IF(X(LMON).LT.XMON)GO TO 4001            00000647
LSEP=IMAX1/2+1                           00000648
4002 LSEP=LSEP+1                          00000649
IF(X(LSEP).LT.XLSEP)GO TO 4002            00000650
CM=1.+#2*M**2                            00000651
IF(ITER/LP*LP.EQ.ITER)PRINT 1,RN          00000652
1 FORMAT(1H0,10X,'BOUNDARY LAYER ANALYSIS FOR REYNOLDS NO. OF ',E10.0,00000653
13,//,5X,'X',9X,                         'M',8X,'DELS',4X,'THETA',3X,'SEP',00000654
210X,'H',9X,'PI',5X,'TAU')                00000655
1000 ISIDE=ISIDE+1                        00000656
SEPMK=0                                    00000657
DO 2 J=ILE,ITE                            00000658
DEL(J)=0.                                  00000659
IF(ISIDE.EQ.2)GO TO 3                      00000660
CP=CPU(J)                                 00000661
4 TEST=(5.*(CM/(1.+#2*CP*M**2)**(.2857143)-1.)) 00000662
EM(J)=0.                                    00000663
IF(TEST.GT.0.)EM(J)=SQRT(TEST)            00000664
DD=1.+#2*EM(J)**2                          00000665
T=CM/DD                                   00000666
UE(J)=EM(J)/M*SQRT(T)                     00000667
GO TO 2                                    00000668
3 CP=CPL(J)                               00000669
GO TO 4                                    00000670
2 CONTINUE
IF(ISIDE.EQ.1)USTR=UE(ITE)                00000671
ILEP1=ILE+1                                00000672
DO 5 J=ILEP1,ITE                           00000673
IF(ISIDE.EQ.2)GO TO 6                      00000674
DSS(J)=SUPPER(J)-SUPPER(J-1)              00000675
GO TO 5                                    00000676
6 DSS(J)=SLOWER(J)-SLOWER(J-1)             00000677
5 DVDS(J)=(UE(J)-UE(J-1))/DSS(J)           00000678
DT=1.                                     00000679
                                         00000680

```

```

SEPR=0 . 00000681
HH=0 . 00000682
IBDS=ITE-1 00000683
DO 200 J=IBDLY,IBDS 00000684
EMT=(EM(J+1)+EM(J))/2. 00000685
UESA=(UE(J+1)+UE(J))/2. 00000686
VM=1.+.2*EMT**2 00000687
T=CM/VM 00000688
RFT=UESA*(T+TR)*T/(1.+TR)*RN 00000689
IF(J.NE.IBDLY)GO TO 30 00000690
THET1=320./RFT 00000691
THET2=THET1 00000692
GE=6.5 00000693
30 FC=1.+.066*EMT**2-.008*EMT**3 00000694
FR=1.-.134*EMT**2+.027*EMT**3 00000695
IND=0 00000696
40 IND=IND+1 00000697
IF(THE1.LT.1.E-06)THET1=1.E-06 00000698
IF(FR.LT.0.)FR=ABS(FR) 00000699
IF(RFT.LT.0.)RFT=ABS(RFT) 00000700
TAU= (FC*(2.4711* ALOG(FR*RFT*THET1)+4.75)+1.5*GE+1724./(GE**2+ 00000701
1200.))-16.87) 00000702
IF(TAU.LT.0.)TAU=-TAU 00000703
TAU=1.*TAU**2 00000704
HB=1./(1.-GE*SQRT(TAU)) 00000705
H=(HB+1.)*(1.+.178*EMT**2)-1. 00000706
SEP=-THET1*DUDS(J+1)/UESA 00000707
IF(SEP.LT.SP)GO TO 41 00000708
IF(X(J+1).LT.XSEP)SEP=SP 00000709
41 PII=H*SEP/TAU 00000710
IF(PII.LT.-1.5)PII=-1.5 00000711
IF(PII.GT.1.E4)PII=1.E4 00000712
50 CONTINUE 00000713
G=6.1*SQRT(PII+1.81)-1.7 00000714
T2=ABS((G-GE)/GE) 00000715
GE=G 00000716
DT2=DT 00000717
DT=(H+2.-EMT**2)*SEP+TAU 00000718
IF(IND.GT.1)GO TO 100 00000719
THET=THET2 00000720

```

```

      THET1=DT*DSS(J+1)+THT          00000721
      THET1=.5*(THET1+THT)          00000722
      GO TO 40                      00000723
100  DT=(DT2+DT)/2.                00000724
      TI=ABS((DT-DT2)/DT)          00000725
      IF(TI.LT.TE2)GO TO 120        00000726
110  THET1=DT*DSS(J+1)+THT          00000727
      THET1=.5*(THET1+THT)          00000728
      IF(IND.LE.500)GO TO 40        00000729
      IF(PII.EQ.-1.5)GO TO 130        00000730
      GO TO 130                      00000731
120  IF(T2.GE.TE1)GO TO 110        00000732
130  THET2=DT*DSS(J+1)+THT          00000733
      THET1=.5*(THET2+THT)          00000734
      SEP=-THET1*DUDS(J+1)/UESA    00000735
      SEPR=(SEPR*DSS(J+1)+SEP*DSS(J))/(DSS(J)+DSS(J+1)) 00000736
      HH=(HH*DSS(J+1)+H*DSS(J))/(DSS(J)+DSS(J+1))    00000737
      DELS=HH*THT                  00000738
      DEL(J)=DELS                  00000739
      IF(DEL(J).GT.0.1)DEL(J)=0.1    00000740
      HS(J)=HH                      00000741
      IF(ITER/LP*LP.EQ.ITER)PRINT 10,X(J),EM(J),DELS,THT,SEPR,HH,PII,TAU 00000742
10  FORMAT(9F10.5,I10,F10.5)        00000743
205 CONTINUE                      00000744
9  CONTINUE                        00000745
      IF(J.EQ.IBDS)GO TO 200        00000746
      SEPR=SEP                      00000747
      HH=H                          00000748
200 CONTINUE                      00000749
      SEPR=SEPR+2.*SEP             00000750
      HH=HS(ITE-1)+(DSS(ITE)/DSS(ITE-1))*(HS(ITE-1)-HS(ITE-2)) 00000751
      HS(ITE)=HH                      00000752
      DELS=HH*THT                  00000753
      DEL(ITE)=DELS                  00000754
      IF(DEL(ITE).GT.0.1)DEL(ITE)=0.1 00000755
      IF(ITER/LP*LP.EQ.ITER)PRINT 10,X(ITE),EM(ITE),DELS,THET2,SEPR, *HH,PII,TAU 00000756
*HH,PII,TAU                      00000757
202 IF(ISIDE.EQ.2)GO TO 203        00000758
      EMSTR=EM(ITE)                  00000759
      HSTR=HH                        00000760

```

```

TSTR=THET2                                00000761
DO 170 J=ILE,IBDS                         00000762
170 IF(DEL(J+1).LT.DEL(J))DEL(J+1)=DEL(J) 00000763
203 CONTINUE                               00000764
IF(ISIDE.EQ.1)GO TO 2200                  00000765
J=ILE                                     00000766
2180 J=J+1                                 00000767
IF(DEL(J+1).LT.DEL(J))GO TO 2185        00000768
IF(J.LT.IBDS)GO TO 2180                  00000769
GO TO 2200                                00000770
2185 IF(X(J).GT.XPC)GO TO 2190          00000771
DEL(J+1)=DEL(J)                           00000772
GO TO 2180                                00000773
2190 J=J+1                                 00000774
IF(J.GT.IBDS)GO TO 2200                  00000775
IF(DEL(J+1).GT.DEL(J))DEL(J+1)=DEL(J)    00000776
IF(J.LT.IBDS)GO TO 2190                  00000776
2200 CONTINUE                             00000777
ISMOTH=2                                  00000778
IF(IMAX.GT.55)ISMOTH=4                  00000779
DO 171 JJ=1,ISMOTH                      00000780
OLD=DEL(ILE)                            00000781
ILEP2=ILE+2                            00000782
DO 171 J=ILEP2,ITE                      00000783
NEW=DEL(J-1)                            00000784
DEL(J-1)=.25*(OLD+NEW+NEW+DEL(J))      00000785
171 OLD=NEW                            00000786
FAC=-DSS(ITE)/DSS(ITE-1)                00000787
DEL(ITE)=FAC*DEL(ITE-2)+(1.-FAC)*DEL(ITE-1) 00000788
DO 172 J=ILEP1,IBDS                     00000789
SLOPE=SLU(J)                           00000790
IF(ISIDE.EQ.2)SLOPE=SLL(J)             00000791
CO=ABS(ATAN(SLOPE))                   00000792
CO=COS(CO)                            00000793
IF(ISIDE.EQ.2)GO TO 173                00000794
DY=DUPOLD(J)+RDEL*(DEL(J)-DUPOLD(J))  00000795
YU(J)=YUORIG(J)+DY/CO                 00000796
DUPOLD(J)=DY                           00000797
GO TO 172                                00000798
173 DY=DLWOLD(J)+RDEL*(DEL(J)-DLWOLD(J)) 00000799

```

4

```
YL(J)=YLORIG(J)-DY/CO          00000800
DLWOLD(J)=DY                   00000801
GO TO 172                      00000802
172 CONTINUE
    SLOPE=SLU(ITE)
    IF(ISIDE.EQ.2)SLOPE=SLL(ITE)
    CO=ABS(ATAN(SLOPE))
    CO=COS(CO)
    IF(ISIDE.EQ.2)GO TO 175
    DY=DUPOLD(ITE)+RDEL*(DEL(ITE)-DUPOLD(ITE))
    YU(ITE)=YUORIG(ITE)+DY/CO
    DUPOLD(ITE)=DY
    GO TO 204
175 DY=DLWOLD(ITE)+RDEL*(DEL(ITE)-DLWOLD(ITE))
    YL(ITE)=YLORIG(ITE)-DY/CO
    DLWOLD(ITE)=DY
204 CONTINUE
    IF(ITEUPC.EQ.0)GO TO 5003
C    **INSERT SEPERATED CORRECTION HERE IF DESIRED**
C    ** SEPERATED COORECTION**
    IF(ISIDE.EQ.2)GO TO 5003
    IF(ICYCLE.GT.1)GO TO 300
    LMON=IMAX/2+1
    CPB=0.
    DO 5001 J=LMON,IBDS
    CPN=CPL(J)
    CPB=AMAX1(CPB,CPN)
5001 CONTINUE
    CPB=0.6
    PRINT 5002,CPB
5002 FORMAT(' ', 'BASE PRESSURE COEFFICIENT = ',F10.3)
    IF(LSEP.EQ.ITE)LSEP=ITE-1
    LSEP1=LSEP+1
    SLOP=(CPB-CPU(LSEP))/(.5-X(LSEP))
    DO 501 J=LSEP1,ITE
501 CPU(J)=SLOP*(X(J)-X(LSEP))+CPU(LSEP)
    ICYCLE=ICYCLE+1
    ISIDE=0
    GO TO 1000
5003 CONTINUE
00000803
00000804
00000805
00000806
00000807
00000808
00000809
00000810
00000811
00000812
00000813
00000814
00000815
00000816
00000817
00000818
00000819
00000820
00000821
00000822
00000823
00000824
00000825
00000826
00000827
00000828
00000829
00000830
00000831
00000832
00000833
00000834
00000835
00000836
00000837
00000838
00000839
```

```

IF(ITELWC.EQ.0)GO TO 300                                00000840
IF(ISIDE.EQ.1)GO TO 300                                00000841
C   ** LOWER SURFACE CORRECTION, IF DESIRED**
IF(ICYBOT.GT.1)GO TO 300                                00000842
IF(LSEP.EQ.ITE)LSEP=ITE-1                               00000843
SLOP=(CPB-CPL(LSEP))/(.5-X(LSEP))                     00000844
LSEP1=LSEP+1                                            00000845
DO 5004 J=LSEP1,ITE                                     00000846
5004 CPL(J)=SLOP*(X(J)-X(LSEP))+CPL(LSEP)            00000847
ICYBOT=ICYBOT+1                                         00000848
ISIDE=1                                                 00000849
GO TO 1000                                              00000850
C   ** END SEPERATED REGION CORRECTION **
300 CONTINUE                                             00000851
IF(ISIDE.LT.2)GO TO 1000                                00000852
XO(1)=-.5                                               00000853
XI(1)=-.5                                               00000854
YI(1)=0.                                                 00000855
NI=ITE-ILE+2                                           00000856
DO 210 I=ILE,ITE                                       00000857
II=I-ILE+2                                             00000858
XI(II)=X(I)                                            00000859
XO(II)=X(I)                                            00000860
210 YI(II)=YU(I)                                         00000861
NO=NI                                                 00000862
CALL ARC(XI,YI,XO,YO,SI,SO,XP,YP,D1Y,D2Y,D3Y,0.0,0.0,1.0,0.0,NI,
      1NO,1)                                              00000863
DO 211 I=ILE,ITE                                       00000864
II=I-ILE+2                                             00000865
YI(II)=YL(I)                                           00000866
IF(I.LT.IBDLY)GO TO 211                               00000867
SLU(I)=YP(II)/XP(II)                                 00000868
211 CONTINUE                                            00000869
IF(ITEUPC.EQ.0)GO TO 5025                            00000870
LSEP1=LSEP+1                                         00000871
IF(XPC.LT.0.495)GO TO 5029                            00000872
DO 5030 J=LSEP1,ITE                                     00000873
IF(SLU(J).GT.0.0)GO TO 5031                            00000874
5030 CONTINUE                                           00000875
GO TO 5025                                              00000876

```

```

5031 DO .5032 I=J,ITE                                00000880
      YU(I)=YU(J-1)                                00000881
5032 SLU(I)=0.0                                     00000882
      GO TO 5025                                     00000883
5029 CONTINUE
      DO 5026 I=LSEP1,ITE                            00000884
          YU(I)=YU(LSEP)+SLU(LSEP)*(X(I)-X(LSEP))  00000885
5026 SLU(I)=SLU(LSEP)                               00000886
5025 CONTINUE
      CALL ARC(XI,YI,X0,Y0,SI,SO,XP,YP,D1Y,D2Y,D3Y,0.0,0.0,-1.0,0.0. 00000887
1NI,NO,1)
      DO 212 I=ILE,ITE                                00000888
          II=I-ILE+2                                  00000889
          YU(I)=ATAN(YU(I)/A1)/PI2                  00000890
          YL(I)=ATAN(YL(I)/A1)/PI2                  00000891
          IF(I.LT.IBDLY)GO TO 212                  00000892
          SLL(I)=YP(II)/XP(II)                      00000893
212 CONTINUE
      IF(ITELWC.EQ.0)GO TO 5027                  00000894
      LSEP1=LSEP+1                                 00000895
      IF(XPC.LT.0.495)GO TO 5036                  00000896
      DO 5033 J=LSEP1,ITE                            00000897
          IF(SLL(J).LT.0.0)GO TO 5034                00000898
5033 CONTINUE
      GO TO 5027                                     00000899
5034 DO 5035 I=J,ITE                                00000900
      YL(I)=YL(J-1)                                00000901
5035 SLL(I)=0.0                                     00000902
      GO TO 5027                                     00000903
5036 CONTINUE
      DO 5028 I=LSEP1,ITE                            00000904
          YL(I)=YL(LSEP)+SLL(LSEP)*(X(I)-X(LSEP))  00000905
5028 SLL(I)=SLL(LSEP)                             00000906
5027 CONTINUE
      HBT=(HSTR+1.)/(1.+.178*EMSTR**2)-1.        00000907
      HBB=(HH+1.)/(1.+.178*EM(ITE)**2)-1.          00000908
      CDF=TSTR*(USTR**(.2.5+.5*HBT))+THET2*(UE(ITE)**(.2.5+.5*HBB)) 00000909
      CDF=2.*CDF                                    00000910
      IF(DPM.LE.CONV) GO TO 5010                  00000911
      IF(ITER.GE.(MITER-1))GO TO 5010              00000912

```

```

GO TO 5008                                00000920
5010 DO 5007 J=ILE,IMAX1                  00000921
      CPU(J)=CPU(J)
5007 CPL(J)=CPLT(J)                      00000922
5008 CONTINUE                            00000923
      RETURN                               00000924
      END                                  00000925
                                         00000926

      SUBROUTINE BDLY                         00000927
C                                         00000928
C **** COMPUTES BOUNDARY LAYER IN THE DESIGN CASE **** 00000929
C                                         00000930
      REAL M,NEW                           00000931
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)00000932
1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),          00000933
1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99)00000934
2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),          00000935
3A1,A2,A12,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,00000936
4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP100000937
5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4                     00000938
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1, 00000939
1JMAX,JCON,JMAX1,NSSP,IW                   00000940
      COMMON/NASH/RN,IBDLY,ITACT,YUORIG(99),YLORIG(99),SUPPER(99),SLOWER00000941
1(99),DEL(99),DUPOLD(99),DLWOLD(99),CDF           00000942
      COMMONADELTA/ITER                      00000943
      COMMON/IPT1/XIBDLY,RDEL,RDELFN,RCPB,SP,XSEP,CONV,CPB,XMON,XLSEP, 00000944
1 MITER,LPI,ITEUPC,ITELWC,XPC             00000945
      DIMENSION UE(99),DSS(99),DUDS(99),YUN(99),YLN(99),EM(99) 00000946
      DIMENSION HS(99)                        00000947
      DIMENSION XI(99),YI(99),X0(99),Y0(99),SI(99),XP(99),DIY(99), 00000948
1D2Y(99),D3Y(99),XIB(99),YIB(99),SO(99),YP(99) 00000949
      ISIDE=0                                00000950
      LSEP=ITE                               00000951
      SEPMK=0                                00000952
      ICYCLE=1                               00000953
      DO 500 J=ILE,ITE                      00000954
      YUN(J)=YU(J)                          00000955
500 YLN(J)=YL(J)                          00000956
      TR=0.3424                            00000957

```

```

`TE1=5.E-03          00000958
TE2=5.E-05          00000959
CM=1.+.2*M**2       00000960
1 FORMAT(1H1,10X,'BOUNDARY LAYER ANALYSIS FOR REYNOLDS NO. OF ',E10.00000961
13,/,5X,'X',9X,'Y',9X,'YNEW',8X,'M',8X,'DELS',4X,'THETA',3X,'SEP',00000962
210X,'H',9X,'PI',5X,'TAU')          00000963
1000 ISIDE=ISIDE+1      00000964
PRINT 1,RN           00000965
DO 2 J=ILE,ITE        00000966
DEL(J)=0.             00000967
IF(ISIDE.EQ.2)GO TO 3 00000968
CP=CPU(J)            00000969
4 TEST=(5.*(CM/(1.+.7*CP*M**2)**(.2857143)-1.)) 00000970
EM(J)=0.              00000971
IF(TEST.GT.0.)EM(J)=SQRT(TEST) 00000972
DD=1.+.2*EM(J)**2     00000973
T=CM/DD               00000974
UE(J)=EM(J)/M*SQRT(T) 00000975
GO TO 2               00000976
3 CP=CPL(J)           00000977
GO TO 4               00000978
2 CONTINUE             00000979
IF(ISIDE.EQ.1)LSTR=UE(ITE) 00000980
ILEP1=ILE+1            00000981
XO(1)=-.5              00000982
XI(1)=-.5              00000983
YI(1)=0.                00000984
NI=ITE-ILE+2            00000985
DO 210 I=ILE,ITE        00000986
II=I-ILE+2              00000987
XI(II)=X(I)             00000988
XO(II)=X(I)             00000989
210 YI(II)=YU(I)         00000990
NO=NI                  00000991
CALL ARC(XI,YI,XO,YO,SI,SO,XP,YP,D1Y,D2Y,D3Y,0.0,0.0,1.0,0.0,NI, 00000992
1NO,1)                  00000993
DO 211 I=ILE,ITE        00000994
II=I-ILE+2              00000995
YI(II)=YL(I)            00000996
SLU(I)=YP(II)/XP(II)    00000997

```

```

211 SUPPER(I)=SO(II) 00000998
    CALL ARC(XI,YI,X0,Y0,SI,SO,XP,YP,D1Y,D2Y,D3Y,0.0,0.0,-1.0,0.0,
1NI,NO,1) 00000999
    DO 212 I=ILE,ITE 00001000
        II=I-ILE+2 00001001
        SLL(I)=YP(II)/XP(II) 00001002
212 SLOWER(I)=SO(II) 00001003
    DO 5 J=ILEP1,ITE 00001004
        IF(ISIDE.EQ.2)GO TO 6 00001005
        DSS(J)=SUPPER(J)-SUPPER(J-1) 00001006
        GO TO 5 00001007
6 DSS(J)=SLOWER(J)-SLOWER(J-1) 00001008
5 DUDS(J)=(UE(J)-UE(J-1))/DSS(J) 00001009
    DT=1.
    SEPR=0.
    HH=0.
    IBDS=ITE-1 00001010
    DO 200 J=IBDLY,IBDS 00001011
        EMT=(EM(J+1)+EM(J))/2. 00001012
        UESA=(UE(J+1)+UE(J))/2. 00001013
        VM=1.+.2*EMT**2 00001014
        T=CM/VM 00001015
        RFT=UESA*(T+TR)*T/(1.+TR)*RN 00001016
        IF(J.NE.IBDLY)GO TO 30 00001017
        THET1=320./RFT 00001018
        THET2=THET1 00001019
        GE=6.5 00001020
30 FC=1.+.066*EMT**2-.008*EMT**3 00001021
    FR=1.-.134*EMT**2+.027*EMT**3 00001022
    IND=0 00001023
40 IND=IND+1 00001024
    TAU=1./((FC*(2.4711* ALOG(FR*RFT*THET1)+4.75)+1.5*GE+1724.)/(GE**2+
1200.))-16.87)**2 00001025
    HB=1./((1.-GE*SQRT(TAU)) 00001026
    H=(HB+1.)*(1.+.178*EMT**2)-1. 00001027
    SEP=-THET1*DUDS(J+1)/UESA 00001028
    IF(SEP.LT.SP)GO TO 41 00001029
    IF(X(J+1).LT.XSEP)SEP=SP 00001030
41 PII=H*SEP/TAU 00001031
    IF(PII.LT.-1.5)PII=-1.5 00001032

```

```

      IF(PII.GT.1.E4)PII=1.E4          00001038
50  CCNTINUE                      00001039
      G=6.1*SQRT(PII+1.81)-1.7      00001040
      T2=ABS((G-GE)/GE)             00001041
      GE=G                          00001042
      DT2=DT                         00001043
      DT=(H+2.-EMT**2)*SEP+TAU       00001044
      IF(IND.GT.1)GO TO 100          00001045
      THT=THET2                      00001046
      THET1=DT*DSS(J+1)+THT         00001047
      THET1=.5*(THET1+THT)           00001048
      GO TO 40                        00001049
100  DT=(DT2+DT)/2.                 00001050
      TI=ABS((DT-DT2)/DT)            00001051
      IF(TI.LT.TE2)GO TO 120          00001052
110  THET1=DT*DSS(J+1)+THT         00001053
      THET1=.5*(THET1+THT)           00001054
      IF(IND.LE.500)GO TO 40          00001055
      IF(PII.EQ.-1.5)GO TO 130        00001056
      PRINT 160                       00001057
160  FORMAT(' PROBLEMS')
      GO TO 130                      00001059
120  IF(T2.GE.TE1)GO TO 110          00001060
130  THET2=DT*DSS(J+1)+THT         00001061
      THET1=.5*(THET2+THT)           00001062
      SEP=-THET1*DUDS(J+1)/UESA      00001063
      SEPR=(SEPR*DSS(J+1)+SEP*DSS(J))/(DSS(J)+DSS(J+1)) 00001064
      HH=(HH*DSS(J+1)+H*DSS(J))/(DSS(J)+DSS(J+1))        00001065
      DELS=HH*THT                     00001066
      DEL(J)=DELS                     00001067
      IF(DEL(J).GT.0.1)DEL(J)=0.1     00001068
      HS(J)=HH                        00001069
      IF(ISIDE.EQ.2)GO TO 8            00001070
      SLOPE=SLU(J)                     00001071
      CO=ABS(ATAN(SLOPE))            00001072
      CO=COS(CO)                      00001073
      YUN(J)=YU(J)-DELS/CO            00001074
      PRINT 10,X(J),YU(J),YUN(J),EM(J),DELS,THT,SEPR,HH,PII, 00001075
      1IND,TAU                         00001076
10  FORMAT(9F10.5,I10,F10.5)          00001077

```

```

IF(SEPMK.EQ.1)GO TO 205          00001078
IF(SEPR.GT.SP)LSEP=J            00001079
IF(LSEP.NE.ITE)SEPMK=1          00001080
205 CONTINUE                      00001081
GO TO 9                          00001082
8 SLOPE=SLL(J)                  00001083
CO=ABS(ATAN(SLOPE))             00001084
CO=COS(CO)                       00001085
YLN(J)=YL(J)+DELS/CO            00001086
PRINT 10,X(J),YL(J),YLN(J),EM(J),DELS,THT,SEPR,HH,PII,
1IND,TAU                         00001087
9 CONTINUE                         00001088
IF(J.EQ.IBDS)GO TO 200           00001089
SEPR=SEP                          00001090
HH=H                            00001091
HH=H                            00001092
200 CONTINUE                      00001093
SEPR=-SEPR+2.*SEP                00001094
HH=HS(ITE-1)+(DSS(ITE)/DSS(ITE-1))*(HS(ITE-1)-HS(ITE-2)) 00001095
HS(ITE)=HH                         00001096
DELS=HH*THET2                     00001097
DEL(ITE)=DELS                      00001098
IF(DEL(ITE).GT.0.1)DEL(ITE)=0.1   00001099
IF(ISIDE.EQ.2)GO TO 201           00001100
SLOPE=SLU(ITE)                   00001101
CO=ABS(ATAN(SLOPE))              00001102
CO=COS(CO)                        00001103
YUN(ITE)=YU(ITE)-DELS/CO          00001104
PRINT 10,X(ITE),YU(ITE),YUN(ITE),EM(ITE),DELS,THET2,SEPR,HH,PII. 00001105
1IND,TAU                         00001106
GO TO 202                         00001107
201 SLOPE=SLL(ITE)                00001108
CO=ABS(ATAN(SLOPE))              00001109
CO=COS(CO)                        00001110
YLN(ITE)=YL(ITE)+DELS/CO          00001111
PRINT 10,X(ITE),YL(ITE),YLN(ITE),EM(ITE),DELS,THET2,SEPR,HH,PII, 00001112
1IND,TAU                         00001113
202 IF(ISIDE.EQ.2)GO TO 203       00001114
EMSTR=EM(ITE)                     00001115
HSTR=HH                           00001116
TSTR=THET2                        00001117

```

```

        DO 170 J=ILE,IBDS                      00001118
170 IF(DEL(J+1).LT.DEL(J))DEL(J+1)=DEL(J) 00001119
203 CONTINUE                                00001120
      IF(ISIDE.EQ.1)GO TO 2200                00001121
      J=ILE                                    00001122
2180 J=J+1                                    00001123
      IF(DEL(J+1).LT.DEL(J))GO TO 2185       00001124
      IF(J.LT.IBDS)GO TO 2180                 00001125
      GO TO 2200                               00001126
2185 IF(X(J).GT.XPC)GO TO 2190              00001127
      DEL(J+1)=DEL(J)                         00001128
      GO TO 2180                             00001129
2190 J=J+1                                    00001130
      IF(DEL(J+1).GT.DEL(J))DEL(J+1)=DEL(J) 00001131
      IF(J.LT.IBDS)GO TO 2190                 00001132
2200 CCNTINUE                                00001133
      ISMOTH=2                                00001134
      IF(IMAX.GT.55)ISMOTH=4                  00001135
      DO 171 JJ=1,ISMOTH                     00001136
      OLD=DEL(ILE)                            00001137
      ILEP2=ILE+2                            00001138
      DO 171 J=ILEP2,ITE                      00001139
      NEW=DEL(J-1)                            00001140
      DEL(J-1)=.25*(OLD+NEW+NEW+DEL(J))     00001141
171 OLD=NEW                                  00001142
      FAC=-DSS(ITE)/DSS(ITE-1)               00001143
      DEL(ITE)=FAC*DEL(ITE-2)+(1.-FAC)*DEL(ITE-1) 00001144
      PRINT 180                                00001145
180 FORMAT(' X          YOLD      YNEW      DELSTAR') 00001146
      DO 172 J=ILEP1,IBDS                     00001147
      SLOPE=SLU(J)                           00001148
      IF(ISIDE.EQ.2)SLOPE=SLL(J)             00001149
      CO=ABS(ATAN(SLOPE))                  00001150
      CO=COS(CO)                            00001151
      IF(ISIDE.EQ.2)GO TO 173                00001152
      YUN(J)=YU(J)-DEL(J)/CO                00001153
      PRINT 174,X(J),YU(J),YUN(J),DEL(J)    00001154
      GO TO 172                             00001155
173 YLN(J)=YL(J)+DEL(J)/CO                 00001156
      PRINT 174,X(J),YL(J),YLN(J),DEL(J)   00001157

```

```

174 FORMAT(4F10.5)          00001158
172 CONTINUE                00001159
    SLOPE=SLU(ITE)           00001160
    IF(ISIDE.EQ.2)SLOPE=SLL(ITE) 00001161
    CO=ABS(ATAN(SLOPE))     00001162
    CO=COS(CO)               00001163
    IF(ISIDE.EQ.2)GO TO 175   00001164
    YUN(ITE)=YU(ITE)-DEL(ITE)/CO 00001165
    PRINT 174,X(ITE),YU(ITE),YUN(ITE),DEL(ITE) 00001166
    GO TO 204               00001167
175 YLN(ITE)=YL(ITE)+DEL(ITE)/CO 00001168
    PRINT 174,X(ITE),YL(ITE),YLN(ITE),DEL(ITE) 00001169
204 IF(ISIDE.LT.2)GO TO 1000 00001170
    HBT=(HSTR+1.)/(1.+.178*EMSTR**2)-1. 00001171
    HBB=(HH+1.)/(1.+.178*EM(ITE)**2)-1. 00001172
    CDF=ISTR*(USTR**(.5+.5*HBT))+THET2*(UE(ITE)**(.5+.5*HBB)) 00001173
    CDF=2.*CDF               00001174
    PRINT 3010,CDF            00001175
3010 FORMAT(1H0,* CDF = ',F10.6) 00001176
    IF(X(LSEP).GE.XSEP)GO TO 3011 00001177
    PRINT 3012,X(LSEP)           00001178
3012 FORMAT(1H ,*UPPER SURFACE SEPARATION DETECTED AT ',F10.5) 00001179
    RETURN
3011 PRINT 3013,XSEP          00001180
3013 FORMAT(1H ,*NO UPPER SURFACE SEPARATION BEFORE',F10.5) 00001182
    RETURN
    END                      00001183
                                00001184

```

SUBROUTINE VALUE	00001185
C	00001186
C **** THIS DETERMINES THE INITIAL SOLUTION *****	00001187
C	00001188
REAL M	00001189
COMMON CPU(99),CPL(99),E(99),DUL(99),DU2(99),DL1(99),DL2(99),D(99)	00001190
1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),	00001191
1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99	0000001192
2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),	00001193
3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,	00001194
4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1	000001195
5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4	00001196
CCMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JBI,	00001197
1JMAX,JCON,JMAX1,NSSP,IW	00001198
COMMON/FINPUT/IREAD	00001199
C INITIALIZES VALUES	00001200
IF(IREAD.EQ.0)GO TO 3	00001201
DO 4 JJ=1,JMAX	00001202
J=JMAX-JJ+1	00001203
READ 5,(P(I,J),I=1,IMAX)	00001204
5 FORMAT(SE15.7)	00001205
4 CONTINUE	00001206
READ5,(PB(I),I=1,IMAX)	00001207
RETURN	00001208
3 CONTINUE	00001209
DO 1 I=1,IMAX	00001210
PB(I)=0.0	00001211
DO 1 J=1,JMAX	00001212
1 P(I,J)=0.0	00001213
RETURN	00001214
END	00001215

```

SUBROUTINE SOLVE(JL,JU)                               00001216
C                                                 00001217
C **** THIS SUBROUTINE SETS UP THE COEFFICIENTS USED IN THE **** 00001218
C RELAXATION SOLUTION **** 00001219
C                                                 00001220
C
REAL M,ML                                         00001221
COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99) 00001222
1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99), 00001223
1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99) 00001224
2,X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99), 00001225
3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2, 00001226
4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1 00001227
5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4, 00001228
COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1, 00001229
1JMAX,JCON,JMAX1,NSSP,IW, 00001230
COMMONADAM/VJB,VJB1,VJBP1, 00001231
COMMONAJS/GG(99),GGP12(99),GGM12(99),GGM32(99),GGP32(99),A3, 00001232
DST2=2.*DS, 00001233
DET2=2.*DE, 00001234
DEDE=DE*DE, 00001235
DSDS=DS*DS, 00001236
DSDE=DS*DE, 00001237
FDS=F/DS, 00001238
DO 3 J=JL,JU, 00001239
IF(I.NE.ILE1)GO TO 300, 00001240
IF(J.NE.JB-1)GO TO 300, 00001241
HLD=P(ILE,JB), 00001242
P(ILE,JB)=PB(ILE), 00001243
300 CONTINUE, 00001244
G=GG(J), 00001245
GP12=GGP12(J), 00001246
GM12=GGM12(J), 00001247
GDE=G/DE, 00001248
U=QI*(COS(ALP)+F*(P(I+1,J)-P1(J))/(2.*DS)), 00001249
V=QI*(SIN(ALP)+G*(P(I,J+1)-P(I,J-1))/(2.*DE)), 00001250
IF(IW.EQ.0)GO TO 6, 00001251
IF(J.EQ.JB)V=V-QI*(G*(CIR/(2.*DE))), 00001252
IF(J.EQ.JB-1)V=V-QI*(G*(CIR/(2.*DE))), 00001253
6 CONTINUE, 00001254
UU=U*U, 00001255

```

```

VV=V*V          00001256
Q=SQRT(UU+VV)  00001257
QQ=Q*Q          00001258
AA=AI2-0.2*(UU+VV-QI2) 00001259
IF(AA.LT.0.0)PRINT 11,I,J 00001260
11 FORMAT(" AA.LT.0.0   I=",I5," J = ",I5) 00001261
IF(AA.LT.0.0)GO TO 21 00001262
IF(IW.EQ.0)GO TO 7 00001263
IF(J.NE.JB.AND.J.NE.JB-1)GO TO 9 00001264
IF(J.EQ.JB-1)GO TO 8 00001265
VVJB=VV          00001266
VJB=V           00001267
AAJB=AA          00001268
QQJB=QQ          00001269
UUJB=UU          00001270
GO TO 7          00001271
8 VVJB1=VV          00001272
VJB1=V           00001273
AAJB1=AA          00001274
QQJB1=QQ          00001275
UUJB1=UU          00001276
GO TO 7          00001277
10 VVJBP1=VV          00001278
VJBP1=V           00001279
QQJBP1=QQ          00001280
AAJBP1=AA          00001281
GO TO 7          00001282
9 IF(J.EQ.JB+1)GO TO 10 00001283
7 CONTINUE          00001284
ML=(UU+VV)/AA          00001285
IF(ML.GT.1.0)GO TO 4 00001286
SUBSONIC POINT      00001287
FAV=1.0*(FP12+FM12) 00001288
FPF=0.5*F*FAV        00001289
SUB(J-1)=(1.-VV/AA)*GDE*GM12/DE 00001290
D(J)=-2.*((1.-UU/AA)*FPF/DSDS/W-(1.-VV/AA)*G*(GP12+GM12)/DEDE 00001291
1-EPS*FDS*U/Q*FM12/DS 00001292
SUP(J)=(1.-VV/AA)*GDE*GP12/DE 00001293
RS(J)=(1.-UU/AA)*F*(-FP12/DSDS*P(I+1,J)+FAV/DSDS*(1.-1./W) 00001294
1*P(I,J)-FM12/DSDS*P(I-1,J))+U*V/AA*FDS*GDE*0.5*(P(I-1,J-1) 00001295
C

```

```

2-P(I-1,J+1)-P(I+1,J-1)+P(I+1,J+1)) 00001296
RS(J)=RS(J)+EPS*FDS*(U/Q*FM12*(-P(I,J)-P(I-1,J)+P1(J))/DS) 00001297
IF(V.LE.0.0)GO TO 200 00001298
SUB(J-1)=SUB(J-1)+EPS*FDS*V/Q*GM12/DE 00001299
D(J)=D(J)-EPS*FDS*V/Q*GM12/DE 00001300
RS(J)=RS(J)+EPS*FDS*V/Q*GM12/DE*(P(I,J-1)-P(I,J)) 00001301
GO TO 3 00001302
200 SUP(J)=SUP(J)-EPS*FDS*V/Q*GP12/DE 00001303
D(J)=D(J)+EPS*FDS*V/Q*GP12/DE 00001304
RS(J)=RS(J)+EPS*FDS*V/Q*GP12/DE*(P(I,J)-P(I,J+1)) 00001305
GO TO 3 00001306
C DAMPING COEFF IS EPSS AT SUPERSONIC POINTS 00001307
C SUPERSONIC CASE, V GT 0 00001308
4 GM32=GGM32(J) 00001309
NSSP=NSSP+1 00001310
GP32=GGP32(J) 00001311
IF(V.LT.0.0)GO TO 5 00001312
SUB(J-1)=UU/QQ*GDE*GM12/DE+EPSS*FDS*V/Q*GM12/DE 00001313
D(J)=-VV/QQ*FDS*FM12/DS-UU/QQ*GDE*(GP12+GM12)/DE-EPSS*FDS* 00001314
1(U/Q*FM12/DS+V/Q*GM12/DE) 00001315
SUP(J)=UU/QQ*GDE*GP12/DE 00001316
RS(J)=-(1.-QQ/AA)*(UU/QQ*F DS*(FM12*(P(I,J)-P1(J))-FM32*(P1(J)- 00001317
1P2(J))/DS+2.*U*V/QQ*F*G*(P(I,J)-P1(J)-P(I,J-1)+P1(J-1))/DSDE 00001318
2+VV/QQ*G*(GM12*(P(I,J)-P(I,J-1))-GM32*(P(I,J-1)-P(I,J-2))/ 00001319
3DEDE)
RS(J)=RS(J)-VV/QQ*F*(FP12*(P(I+1,J)-P(I,J))+FM12*P(I-1,J))/DSDS 00001321
1+U*V/QQ*F*G*(P(I-1,J-1)-P(I-1,J+1)-P(I+1,J-1)+P(I+1,J+1))/DSDE 00001322
2*0.5 00001323
3+EPSS*F*(U/Q*FM12*(-P(I,J)-P(I-1,J)+P1(J))/DSDS+V/Q*GM12*(P(I,J-1) 00001324
4-P(I,J))/DSDE) 00001325
GO TO 3 00001326
C SUPERSONIC CASE V LT 0 00001327
5 SUB(J-1)=UU/QQ*G DE*GM12/DE 00001328
D(J)=-VV/QQ*F DS*FM12/DS-UU/QQ*G DE*(GM12+GP12)/DE-EPSS*F DS* 00001329
1(U/Q*FM12/DS-V/Q*GP12/DE) 00001330
SUP(J)=UU/QQ*GDE*GP12/DE-EPSS*FDS*V/Q*GP12/DE 00001331
RS(J)=-(1.-QQ/AA)*(UU/QQ*F DS*(FM12*(P(I,J)-P1(J))-FM32*(P1(J)- 00001332
1P2(J))/DS+2.*U*V/QQ*F*G*(-P(I,J)+P1(J)+P(I,J+1)-P1(J+1))/DSDE 00001333
2+VV/QQ*G*(GP12*(P(I,J)-P(I,J+1))-GP32*(P(I,J+1)-P(I,J+2))/ 00001334
3DEDE) 00001335

```

```

RS(J)=RS(J)-VV/QQ*F*(FP12*(P(I+1,J)-P(I,J))+FM12*P(I-1,J))/DSDS 00001336
1+U*V/QQ*F*G*(P(I-1,J-1)-P(I-1,J+1)-P(I+1,J-1)+P(I+1,J+1))/DSDE 00001337
1*0.5 00001338
2+EPSS*F*(U/Q*FM12*(-P(I,J)-P(I-1,J)+P1(JJ))/DSDS+V/Q*GP12*(P(I,J)-
3P(I,J+1))/DSDE) 00001339
IF(I.NE.ILE1)GO TO 3 00001340
IF(J.NE.JB-1)GO TO 3 00001342
P(ILE,JB)=HLD 00001343
3 CONTINUE 00001344
RETURN 00001345
21 DO 18 JJ=1,JMAX 00001346
J=JMAX+1-JJ 00001347
PRINT 19,J 00001348
19 FORMAT(1H , 'ROW ',I5) 00001349
PRINT 20,(P(I,J),I=1,IMAX) 00001350
20 FORMAT(1H ,10E11.3) 00001351
18 CONTINUE ) 00001352
PRINT 19,JB 00001353
PRINT 20,(PB(I),I=1,IMAX) 00001354
STOP 00001355
END 00001356

```

```

SUBROUTINE PRESS 00001357
C 00001358
C ***** THIS COMPUTES THE CP DISTRIBUTION ON THE AIRFOIL ***** 00001359
C 00001360
REAL M 00001361
COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99) 00001362
1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99), 00001363
1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99) 0000001364
2),X(99),Y(99),YL(99),YL(99),SLU(99),SLL(99), 00001365
3A1,A2,A12,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,Q12, 00001366
4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBPI,QQJBPI,AAJBP 00001367
5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4 00001368
COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IM,X1,INV,JB,JA1,JB1, 00001369
1JMAX,JCON,JMAX1,NSSP,IW 00001370
COMMON/FIX/MHALF 00001371

```

```

COMMON/NASH/RN,IBDLY,ITACT,YUORIG(99),YLORIG(99),SUPPER(99),SLOWER00001372
1(99),DEL(99),DUPOLD(99),DLWOLD(99),CDF 00001373
C COMPUTES CP ON AIRFOIL 00001374
IPRM=0 00001375
IF(ITACT.EQ.1)IPRM=1 00001376
19 IPRM=IPRM+1 00001377
IEND=IMAX1 00001378
JB2=JB-2 00001379
DO 1 I=ILE,IEND 00001380
TEMP2=YU(I) 00001381
IF(I.GT.ITE)YU(I)=0.0001 00001382
DO 2 J=JB2,JMAX1 00001383
IF(YU(I).GT.E(J).AND.YU(I).LE.E(J+1))GO TO 3 00001384
2 CONTINUE 00001385
3 JA=J+1 00001386
IF(JA.LE.JB)JA=JB+1 00001387
F=FF(I) 00001388
IF(IPRM.EQ.2)GO TO 20 00001389
IF(I.GT.(ILE+1))GO TO 15 00001390
20 U=0.0 00001391
U=QI*((COS(ALP)+F*((P(I+1,JA-1)-P(I-1,JA-1))/(2.*DS)+(YU(I)-E(JA-1) 00001392
1)*(P(I+1,JA)-P(I+1,JA-1)-P(I-1,JA)+P(I-1,JA-1))/(2.*DS*DE))) 00001393
GO TO 16 00001394
15 CONTINUE 00001395
C U USING BACKWARD DIFFERENCE ON PHI X 00001396
U=QI*((COS(ALP)+F*((3.*P(I,JA-1)-4.*P(I-1,JA-1)+P(I-2,JA-1))/(2.*DS 00001397
1)+(YU(I)-E(JA-1) 00001398
1)*(P(I+1,JA)-P(I+1,JA-1)-P(I-1,JA)+P(I-1,JA-1))/(2.*DS*DE))) 00001399
16 CONTINUE 00001400
UU=U*U 00001401
GB=A11/(1.+TAN(P12*YU(I))**2) 00001402
100 V=QI*((SIN(ALP)+GB*(-3.*P(I,JA-1)+4.*P(I,JA)-P(I,JA+1))/(2.*DE) 00001403
1+(YU(I)-E(JA-1))*(P(I,JA-1)-2.*P(I,JA)+P(I,JA+1))/(DE**2))) 00001404
101 VV=V*V 00001405
YU(I)=TEMP2 00001406
1 CPU(I)=(1./(0.7*M*M))*((1.+0.2*M*M*(1.-(UU+VV)/Q12))**3.5-1.) 00001407
ITE1=ILE+1 00001408
DO 4 I=ILE1,IMAX 00001409
TEMP1=PB(I) 00001410
PB(I)=P(I,JB) 00001411

```

```

~ 4 P(I,JB)=TEMP1          00001412
    JB2=JB+2
    DO 5 I=ILE,IEND         00001413
    TEMP2=YL(I)
    IF(I.GT.ITE)YL(I)=-0.0001 00001414
    DO 6 JJ=1,JB             00001415
    J=JB2-JJ
    IF(YL(I).GE.E(J).AND.YL(I).LT.E(J+1))GO TO 7 00001416
    6 CONTINUE                00001417
    7 JA=J                   00001418
    IF(JA.GE.JB)JA=JB-1      00001419
    F=FF(I)
    IF(IPRM.EQ.2)GO TO 21    00001420
    IF(I.GT.(ILE+1))GO TO 17 00001421
21   U=0.0                   00001422
    U=QI*((COS(ALP)+F*((P(I+1,JA+1)-P(I-1,JA+1))/(2.*DS)+(YL(I)-E(JA+1) 00001423
    1)*(P(I+1,JA+1)-P(I+1,JA)-P(I-1,JA+1)+P(I-1,JA))/(2.*DS*DE))) 00001424
    GO TO 18                00001425
    17 CONTINUE                00001426
C     U USING BACKWARDS DIFFERNCE SCHEME ON PHI X          00001427
    U=QI*((COS(ALP)+F*((3.*P(I,JA+1)-4.*P(I-1,JA+1)+P(I-2,JA+1))/(2.*DS 00001428
    1)+(YL(I)-E(JA+1) 00001429
    1)*(P(I+1,JA+1)-P(I+1,JA)-P(I-1,JA+1)+P(I-1,JA))/(2.*DS*DE))) 00001430
    18 CONTINUE                00001431
    UU=U*U                  00001432
    GB=A11/(1.+TAN(YL(I)*PI2)**2) 00001433
102   V=QI*((SIN(ALP)+GB*((3.*P(I,JA+1)-4.*P(I,JA)+P(I,JA-1))/(2.*DE)+ 00001434
    1*(YL(I)-E(JA+1))*(P(I,JA+1)-2.*P(I,JA)+P(I,JA-1)/(DE**2))) 00001435
    103  VV=V*V                00001436
    YL(I)=TEMP2              00001437
    5 CPL(I)=(1./(0.7*M*M))*((1.+0.2*M*M*(1.-(UU+VV)/QI2))**3.5-1.) 00001438
    DO 8 I=ILE1,IMAX          00001439
    TEMP1=PB(I)
    PB(I)=P(I,JB)            00001440
    8 P(I,JB)=TEMP1          00001441
    IF(ITACT.EQ.1)RETURN      00001442
11   IF(IPRM.EQ.1)PRINT 200 00001443
200 FORMAT(1H , "CP BY BACKWARD DIFFERENCES") 00001444
    IF(IPRM.EQ.2)PRINT 201 00001445
201 FORMAT(1H , "CP BY CENTRAL DIFFERENCES") 00001446

```

```

PRINT 12                                00001452
12 FORMAT(1H , 'X',10X,'CPU',10X,'CPL')
PRINT 13,(X(I),CPU(I),CPL(I),I=ILE,I MAX1) 00001453
13 FORMAT(1H ,3F10.3)                      00001454
IF(IPRM.LT.2)GO TO 19                    00001455
RETURN                                     00001456
END                                         00001457
                                           00001458

SUBROUTINE FLOW1                         00001459
C                                         00001460
C ****SOLVES FLOW IN FRONT OF THE AIRFOIL **** 00001461
C                                         00001462
REAL M                                    00001463
CCMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99) 00001464
1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99), 00001465
1PI(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99) 00001466
2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99), 00001467
3AI,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2, 00001468
4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJB1,QQJB1,AAJBP1 00001469
5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4 00001470
CCMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,I MAX,I MAX1,INV,JB,JA1,JBI, 00001471
1JMAX,JCON,J MAX1,NSSP,IW 00001472
C RELAXES FLOW IN FRONT OF AIRFOIL 00001473
NSSP=0 00001474
ISTAR=2 00001475
IF(M.LT.1.0)GO TO 1 00001476
ISTAR=3 00001477
DO 2 J=1,JMAX 00001478
P1(J)=P(2,J) 00001479
2 P2(J)=P(1,J) 00001480
1 CGNTINUE 00001481
AI=1117.0 00001482
AI2=AI**2 00001483
QI=M*AI 00001484
QI2=QI**2 00001485
DO 1000 I=ISTAR,ILE1 00001486
F=FF(I) 00001487
FP12=FFP12(I) 00001488

```

```

FM12=FFM12(I)                                00001489
FM32=FFM32(I)                                00001490
CALL SOLVE(2,JMAX1)                           00001491
RS(2)=RS(2)-SUB(1)*P(I,1)                     00001492
RS(JMAX1)=RS(JMAX1)-SUP(JMAX1)*P(I,JMAX)    00001493
CALL TRID(2,JMAX1)                           00001494
DO 6 J=2,JMAX1                               00001495
DP=ABS(P(I,J)-RS(J))                         00001496
IF(DP.GT.DPM)ICON=I                          00001497
IF(DP.GT.DPM)JCON=J                          00001498
IF(DP.GT.DPM)DPM=DP                         00001499
P2(J)=P1(J)                                 00001500
P1(J)=P(I,J)                               00001501
6 P(I,J)=RS(J)                                00001502
P2(1)=P1(1)                                 00001503
P2(JMAX)=P1(JMAX)                           00001504
P1(1)=P(I,1)                                 00001505
P1(JMAX)=P(I,JMAX)                           00001506
1000 CONTINUE                                00001507
RETURN                                     00001508
END                                         00001509

```

```

SUBROUTINE COORD                                00001510
C                                              00001511
C *****      SETS UP COORDINATES IN COMPUTATIONAL AND PHYSICAL GRIDS *00001512
C                                              00001513
REAL M                                         00001514
COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)00001515
1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),                                00001516
1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(9900001517
2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),                                00001518
3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,00001519
4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJP1,QQJP1,AAJP100001520
5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4                                00001521
COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1, 00001522
1JMAX,JCON,JMAX1,NSSP,IW                                00001523
COMMON/FIX/MHALF                                00001524
COMMON/J5/GG(99),GGP12(99),GGM12(99),GGM32(99),GGP32(99),A3 00001525

```

```

DE=2.0*(JMAX-1)
IF(INV.EQ.0)GO TO 999
READ 997,X1,X2
997 FORMAT(2F10.5)
999 CONTINUE
DS=2.*{(1.+S4)/(IMAX-1)}
C THIS PROGRAM DEPENDS UPON TRANSFORMATION USED
S(I)=-1.0-S4
E(I)=-1.0
S(IMAX)=1.0+S4
E(JMAX)=1.0
IMAX1=IMAX-1
JMAX1=JMAX-1
DO 2 I=2,IMAX1
2 S(I)=S(I-1)+DS
S3=-S4+0.5*DS-0.01
DO 11 I=1,IMAX1
IF(S3.GE.S(I).AND.S3.LT.S(I+1))GO TO 12
11 CONTINUE
12 I3=I+1
I31=I3-1
IM=IMAX/2+1
I4=IMAX-I31
I41=I4+1
DO 13 I=2,I31
13 X(I)=-X4+A2*TAN(PI2*(S(I)+S4))+A3*TAN(PI2*(S(I)+S4)**3)
TERM1=.5*X4/S4-.25*PI*A2
TERM2=(.5*PI*A2*S4-X4)/(2.*S4**3)
DO 14 I=I3,I4
14 X(I)=S(I)*(TERM1+TERM2*S(I)**2)
DO 16 I=I41,IMAX1
16 X(I)=X4+A2*TAN(PI2*(S(I)-S4))+A3*TAN(PI2*(S(I)-S4)**3)
DO 3 J=2,JMAX1
E(J)=E(J-1)+DE
3 Y(J)=A1*TAN(PI2*E(J))
PRINT 4
4 FORMAT(//,25X,'X-Y GRID SYSTEM',//)
PRINT 5,(I,X(I),I=2,IMAX1)
5 FORMAT( 6(I5,E12.4))
PRINT 5,(J,Y(J),J=2,JMAX1)

```

09

```

    DO 7 I=2,IMAX1                                00001566
    IF(X1.GE.X(I).AND.X1.LT.X(I+1))GO TO 8      00001567
7 CONTINUE                                         00001568
8 I1=I+1                                           00001569
   SLE=-0.5                                         00001570
   DO 9 I=2,IMAX1                                00001571
   IF(SLE.GE.X(I).AND.SLE.LE.X(I+1))GO TO 10     00001572
9 CONTINUE                                         00001573
10 ILE=I+1                                         00001574
   QUAN1=S(2)+S4                                  00001575
   FF(2)= PI2*A2*(1.+TAN(PI2*QUAN1)**2)+1.5*A3*(1.+TAN(PI2*QUAN1)**3)00001576
1)**2)*(QUAN1**2)*PI                            00001577
   FF(2)=1./FF(2)                                 00001578
   QUAN1=S(2)+0.5*DS+S4                          00001579
   FFP12(2)=PI2*A2*(1.+TAN(PI2*QUAN1)**2)+1.5*A3*(1.+TAN(PI2*QUAN1)**3)00001580
1)**2)*(QUAN1**2)*PI                            00001581
   FFP12(2)=1./FFP12(2)                           00001582
   QUAN1=S(2)-0.5*DS+S4                          00001583
   FFM12(2)=PI2*A2*(1.+TAN(PI2*QUAN1)**2)+1.5*A3*(1.+TAN(PI2*QUAN1)**3)00001584
1)**2)*(QUAN1**2)*PI                            00001585
   FFM12(2)=1./FFM12(2)                           00001586
   FFM1(2)=0.0                                     00001587
   DO 18 I=3,I31                                  00001588
   QUAN1=S(I)+S4                                  00001589
   FF(I)= PI2*A2*(1.+TAN(PI2*QUAN1)**2)+1.5*A3*(1.+TAN(PI2*QUAN1)**3)00001590
1)**2)*(QUAN1**2)*PI                            00001591
   FF(I)=1./FF(I)                                 00001592
   QUAN1=S(I)+0.5*DS+S4                          00001593
   FFP12(I)=PI2*A2*(1.+TAN(PI2*QUAN1)**2)+1.5*A3*(1.+TAN(PI2*QUAN1)**3)00001594
1)**2)*(QUAN1**2)*PI                            00001595
   FFP12(I)=1./FFP12(I)                           00001596
   FFM12(I)=FFP12(I-1)                           00001597
   FFM1(I)=FF(I-1)                               00001598
18 FFM32(I)=FFM12(I-1)                           00001599
   FFP12(L31)=1./(TERM1+3.*TERM2*(S(I31)+0.5*DS)**2) 00001600
   IM1=IM-1                                       00001601
   DO 19 I=I3,IM                                  00001602
   FF(I)=1./(TERM1+3.*TERM2*S(I)**2)              00001603
   FFP12(L)=1./(TERM1+3.*TERM2*(S(I)+0.5*DS) **2) 00001604
   FFM12(I)=FFP12(I-1)                           00001605

```

```

    FFM1(I)=FF(I-1)                                00001606
19 FFM32(I)=FFM12(I-1)                            00001607
    FFP12(IM)=FFM12(IM)                            00001608
    DO 800 II=2,IM1                                00001609
    I=IMAX-II+1                                    00001610
    FF(I)=FF(II)                                    00001611
    FFP12(I)=FFM12(II)                            00001612
    FFM12(I)=FFP12(II)                            00001613
    FFM1(I)=FF(II+1)                                00001614
800 FFM32(I)=FFP12(II+1)                            00001615
    GGP32(2)=A11*COS(PI2*(E(3)+0.5*DE))**2      00001616
    GG(2)=A11*COS(PI2*E(2))**2                  00001617
    GGP12(2)=A11*COS(PI2*(E(2)+0.5*DE))**2      00001618
    GGM12(2)=A11*COS(PI2*(E(2)-0.5*DE))**2      00001619
    GGM32(2)=0.0                                    00001620
    DO 801 J=3,JMAX1                               00001621
    IF(J.EQ.JMAX1) GO TO 804                      00001622
    GGP32(J)=A11*COS(PI2*(E(J)+1.5*DE))**2      00001623
804 GG(J)=A11*COS(PI2*E(J))**2                  00001624
    GGP12(J)=A11*COS(PI2*(E(J)+0.5*DE))**2      00001625
    GGM12(J)=GGP12(J-1)                            00001626
801 GGM32(J)=GGM12(J-1)                            00001627
    GGP32(JMAX1)=0.0                                00001628
    RETURN
    END

```

```

SUBROUTINE FLOW3                                00001631
C                                               00001632
C ***** SOLVES FOW IN THE INVERSE REGION *****
C                                               00001633
C                                               00001634
REAL M                                         00001635
COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)00001636
1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),          00001637
1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(9900001638
2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),          00001639
3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,00001640
4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,QQJB,          00001641
5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4                00001642

```

१८

```

C CCOMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1, 00001643
C 1JMAX,JCON,JMAX1,NSSP,IW 00001644
C CCOMMON/BAKER/TEMP3,TEMP4 00001645
C COMMON/DELTA/ITER 00001646
C COMMON/AFIX/MHALF 00001647
C RELAXES FLOW IN INVERSE REGION 00001648
C JA=JA1 00001649
C IS=I1-2 00001650
C DO 14 I=IS,IMAX 00001651
C 14 TEMP(I)=P(I,JB-1) 00001652
C DO 3000 I=I1,IMAX1 00001653
C IF(X(I).GE.X2)GO TO 3100 00001654
C FLOW ABOVE AIRFOIL 00001655
C JB2=JB-2 00001656
C DO 3 J=JB2,JMAX1 00001657
C IF(YU(I).GT.E(J).AND.YU(I).LE.E(J+1))GO TO 4 00001658
C 3 CONTINUE 00001659
C 4 JA=J+1 00001660
C IF(JA.LE.JB)JA=JB+1 00001661
C F=FF(I) 00001662
C FP12=FFP12(I) 00001663
C FM12=FFM12(I) 00001664
C FM1=FFM1(I) 00001665
C FM32=FFM32(I) 00001666
C GB=A11/(1.+TAN(YU(I))*PI2)**2 00001667
C VV=SLU(I)**2 00001668
C UU=1.0 00001669
C 4000 DUI(I)=-COS(ALP)+((1. / (1.+VV/UU))*(1.-((1.+0.7*M*M*CPU(I)))**0.2800001670
C 16-1.)*5. / (M*M))**0.5-F*((-4.*P1(JA-1)+P2(JA-1))/(2.*DS)) 00001671
C 2-F*(YU(I)-E(JA-1))*(P(I+1,JA)-P(I+1,JA-1)-P1(JA)+P1(JA-1))/(2.* 00001672
C 3*DS*DE) 00001673
C DUI(I)=DUI(I)/(F*(1.5/DS)) 00001674
C 4001 CONTINUE 00001675
C DU2(I)=-P(I,JA)+2.0*DUI(I) 00001676
C P(I,JA-1)=DUI(I) 00001677
C P(I,JA-2)=DU2(I) 00001678
C CALL SOLVE(JA,JMAX1) 00001679
C RS(JA)=RS(JA)-SUB(JA-1)*P(I,JA-1) 00001680
C RS(JMAX1)=RS(JMAX1)-SUP(JMAX1)*P(I,JMAX) 00001681
C CALL TRID(JA,JMAX1) 00001682

```

```

DO 1 J=JA,JMAX1                               00001683
DP=ABS(P(I,J)-RS(J))                         00001684
IF(DP.GT.DPM)ICON=I                           00001685
IF(DP.GT.DPM)JCON=J                           00001686
IF(DP.GT.DPM)DPM=DP                           00001687
P2(J)=P1(J)                                 00001688
P1(J)=P(I,J)                                00001689
1 P(I,J)=RS(J)                               00001690
JAM1=JA-1                                  00001691
DO 2 J=JB,JAM1                               00001692
P2(J)=P1(J)                                 00001693
2 P1(J)=P(I,J)                               00001694
VV=SLU(I)**2                                00001695
UU=1.0                                     00001696
4002 DU1(I)=-COS(ALP)+((1./(1.+VV/UU))*(1.-((1.+0.7*M*M*CPU(I )))**0.2800001697
16-1.)*5./(M*M))**0.5-F*((-4.*P(I-1,JA-1)+P(I-2,JA-1))/(2.*DS)) 00001698
2-F*(YU(I)-E(JA-1))*(P(I+1,JA)-P(I+1,JA-1)-P(I-1,JA)+P(I-1,JA-1))/ 00001699
3(2.*DS*DE)
DU1(I)=DU1(I)/(F*(1.5/DS))                00001700
4003 CONTINUE                                00001701
DU2(I)=-P(I,JA)+2.0*DU1(I)                  00001702
P(I,JA-1)=DU1(I)                            00001703
P(I,JA-2)=DU2(I)                            00001704
P2(JMAX)=P1(JMAX)                           00001705
P1(JMAX)=P(I,JMAX)                           00001706
JA1=JA                                    00001707
3000 CONTINUE                                00001708
3100 JA=JB1                                 00001709
DO 5 I=IS,IMAX                               00001710
P(I,JB-1)=TEMP(I)                           00001711
TEMP(I)=P(I,JB+1)                           00001712
TEMP1=PB(I)                                 00001713
PB(I)=P(I,JB)                               00001714
5 P(I,JB)=TEMP1                             00001715
6 TEMP1=TEMP4                               00001716
TEMP4=P1(JB)                                00001717
P1(JB)=TEMP1                               00001718
TEMP1=TEMP3                                00001719
TEMP3=P2(JB)                                00001720
P2(JB)=TEMP1                               00001721

```

```

DO 3500 I=II,IMAX1          00001723
C   FLOW BELOW AIRFOIL      00001724
IF(X(I).GE.X2)GO TO 3600    00001725
JB2=JB+2                   00001726
DO 10 JJ=1,JB               00001727
J=JB2-JJ                   00001728
IF(YL(I).GE.E(J).AND.YL(I).LT.E(J+1))GO TO 11 00001729
10 CONTINUE                  00001730
11 JA=J                     00001731
IF(JA.GE.JB)JA=JB-1        00001732
F=FF(I)                     00001733
FP12=FFP12(I)               00001734
FM12=FFM12(I)               00001735
FM1=FFM1(I)                 00001736
FM32=FFM32(I)               00001737
GB=A11/(1.+TAN(YL(I))*PI2)**2 00001738
VV=SLL(I)**2                00001739
UU=1.0                      00001740
4004 DL1(I)=-COS(ALP)+((1./(1.+VV/UU))*(1.-((1.+0.7*M*M*CPL(I ))**0.2800001741
16-1.)*5./ (M*M))**0.5-F*((-4.*P1(JA+1)+P2(JA+1))/(2.*DS)) 00001742
2-F*(YL(I)-E(JA+1))*(P(I+1,JA+1)-P(I+1,JA)-P1(JA+1)+P1(JA))/(2. 00001743
3*DS*DE)
DL1(I)=DL1(I)/(F*(1.5/DS)) 00001744
4005 CONTINUE                00001745
DL2(I)=2.*DL1(I)-P(I,JA)    00001746
P(I,JA+1)=DL1(I)            00001747
P(I,JA+2)=DL2(I)            00001748
CALL SOLVE(2,JA)             00001749
RS(2)=RS(2)-SUB(1)*P(I,1)    00001750
RS(JA)=RS(JA)-SUP(JA)*P(I,JA+1) 00001751
CALL TRID(2,JA)              00001752
DO 7 J=2,JA                  00001753
DP=ABS(P(I,J)-RS(J))        00001754
IF(DP.LE.DPM)GO TO 8         00001755
ICON=I                       00001756
JCON=J                       00001757
DPM=DP                       00001758
8 P2(J)=P1(J)                00001759
P1(J)=P(I,J)                 00001760
7 P(I,J)=RS(J)               00001761

```

```

JAM1=JA+1                                00001763
DO 9 J=JAM1,JB                            00001764
P2(J)=P1(J)                                00001765
9 P1(J)=P(I,J)                            00001766
VV=SLL(I)**2                                00001767
UU=1.0                                    00001768
4006 DL1(I)=-COS(ALP)+((1.+(1.+VV/UU))*(1.-((1.+0.7*M*M*CPL(I )))**0.2800001769
16-1.)*5./({M*M}))**0.5-F*((-4.*P(I-1,JA+1)+P(I-2,JA+1))/(2.*DS)) 00001770
2-F*(YL(I)-E(JA+1))*(P(I+1,JA+1)-P(I+1,JA)-P(I-1,JA+1)+P(I-1,JA))/ 00001771
3(2.*DS*DE)                                00001772
DL1(I)=DL1(I)/{F*(1.5/DS)}                00001773
4007 CONTINUE                                00001774
DL2(I)=2.*DL1(I)-P(I,JA)                  00001775
P(I,JA+1)=DL1(I)                            00001776
P(I,JA+2)=DL2(I)                            00001777
P2(1)=P1(1)                                00001778
P1(1)=P(I,1)                                00001779
JB1=JA                                    00001780
3500 CONTINUE                                00001781
3600 DO 12 I=IS,IMAX                      00001782
P(I,JB+1)=TEMP(I)                            00001783
TEMP1=PB(I)                                00001784
PB(I)=P(I,JB)                                00001785
12 P(I,JB)=TEMP1                            00001786
P1(JB)=TEMP4                                00001787
P2(JB)=TEMP3                                00001788
RETURN                                     00001789
END                                         00001790

```

```

SUBROUTINE FLOW2                                00001701
C                                              0000175
C ***** SOLVES FLOW IN THE DIRECT REGION ***** 00001793
C                                              00001794
C                                              00001795
      REAL M                                     00001796
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99) 00001796
      1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99), 00001797
      1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(9900001798
      2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99), 00001799
      3A1,A2,A12,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,00001800
      4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJB1,QQJBP1,AAJBP100001801
      5,Q,QQ,UUJB1,PI,P12,A22,A11,X4,S4 00001802
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1, 00001803
      IJMAX,JCON,JMAX1,NSSP,IW 00001804
      COMMON/BAKER/TEMP3,TEMP4 00001805
C RELAXES FLOW ABOVE AND BELOW AIRFOIL 00001806
      JB=JMAX/2+1 00001807
      JB1=JB-1 00001808
      JA1=JB+1 00001809
      JB2=JB-2 00001810
      TEMP5=P1(JB) 00001811
      TEMP6=P2(JB) 00001812
      DO 12 I=ILE,I1 00001813
12 TEMP(I)=P(I,JB-1) 00001814
      IF(ILE.GT.I11)GO TO 2001 00001815
      DO 2000 I=ILE,I11 00001816
C FLOW ABOVE AIRFOIL 00001817
      F=FF(I) 00001818
      FP12=FFP12(I) 00001819
      FM12=FFM12(I) 00001820
      FM32=FFM32(I) 00001821
      GB=A11/(1.+TAN(PI2*YU(I))**2) 00001822
      DO 1 J=JB2,JMAX1 00001823
      IF(YU(I).GT.E(J).AND.YU(I).LE.E(J+1))GO TO 2 00001824
1 CONTINUE 00001825
2 JA=J+1 00001826
      IF(JA.LE.JB)JA=JB+1 00001827
      DU1(I)=SLU(I)*(COS(ALP)+F*((P(I+1,JA-1)-P1(JA-1))/(2.*DS) 00001828
      1+(YU(I)-E(JA-1))*(P(I+1,JA)-P(I+1,JA-1)-P1(JA)+P1(JA-1))/(2.*DS*DE00001829
      _2))-SIN(ALP)-GB*((4.*P(I,JA)-P(I,JA+1))/(2.*DE)+(YU(I)-E(JA-1))* 00001830

```

```

3(-2.*P(I,JA)+P(I,JA+1))/(DE**2))          00001831
DU1(I)=DU1(I)/(-1.5*GB/DE      +(YU(I)-E(JA-1))/(DE**2)*GB) 00001832
DU2(I)=-P(I,JA)+DU1(I)*2.0                  00001833
P(I,JA-1)=DU1(I)                           00001834
P(I,JA-2)=DU2(I)                           00001835
CALL SOLVE(JA,JMAX1)                      00001836
RS(JA)=RS(JA)-SUB(JA-1)*P(I,JA-1)          00001837
RS(JMAX1)=RS(JMAX1)-SUP(JMAX1)*P(I,JMAX)  00001838
CALL TRID(JA,JMAX1)                       00001839
DO 4 J=JA,JMAX1                           00001840
DP=ABS(P(I,J)-RS(J))                     00001841
IF(DP.GT.DPM)ICON=I                      00001842
IF(DP.GT.DPM)JCON=J                      00001843
IF(DP.GT.DPM)DPM=DP                      00001844
P2(J)=P1(J)                            00001845
P1(J)=P(I,J)                           00001846
4 P(I,J)=RS(J)                           00001847
JAM1=JA-1                             00001848
DO 5 J=JB,JAM1                          00001849
P2(J)=P1(J)                           00001850
5 P1(J)=P(I,J)                           00001851
P2(JMAX)=P1(JMAX)                      00001852
P1(JMAX)=P(I,JMAX)                      00001853
JA1=JA                                00001854
DU1(I)=SLU(I)*(COS(ALP)+F*((P(I+1,JA-1)-P(I-1,JA-1))/(2.*DS)
1+(YU(I)-E(JA-1))*(P(I+1,JA)-P(I+1,JA-1)-P(I-1,JA)+P(I-1,JA-1))/(
2*(2.*DS*DE)))-SIN(ALP)-GB*((4.*P(I,JA)-P(I,JA+1))/(2.*DE)+ 00001856
3*(YU(I)-E(JA-1))*(-2.*P(I,JA)+P(I,JA+1))/(DE**2))           00001857
DU1(I)=DU1(I)/(-1.5*GB/DE+(YU(I)-E(JA-1))/(DE**2)*GB)        00001858
DU2(I)=-P(I,JA)+DU1(I)*2.0              00001859
P(I,JA-1)=DU1(I)                      00001860
P(I,JA-2)=DU2(I)                      00001861
2000 CONTINUE                           00001862
2001 CONTINUE                           00001863
TEMP3=P2(JB)                           00001864
TEMP4=P1(JB)                           00001865
TEMP3=P2(JB)                           00001866
G FLOW BELOW AIRFOIL                   00001867
DO 8 I=ILE,I1                         00001868
P(I,JB-1)=TEMP(I)                      00001869
TEMP(I)=P(I,JB+1)                      00001870

```

```

      TEMP1=PB(I)          0000187'
      PB(I)=P(I,JB)        00001872
      P(I,JB)=TEMP1        00001873
  8 CONTINUE                00001874
      P1(JB)=TEMP5        00001875
      P2(JB)=TEMP6        00001876
      IF(ILE.GT.I11)GO TO 2501 00001877
      DO 2500 I=ILE,I11      00001878
      GB=A11/(1.+TAN(P12*YL(I))**2) 00001879
      F=FF(I)                00001880
      FP12=FFP12(I)          00001881
      FM12=FFM12(I)          00001882
      FM32=FFM32(I)          00001883
      DO 6 JJ=1,JMAX1        00001884
      J=JB-JJ+2              00001885
      IF(YL(I).GE.E(J).AND.YL(I).LT.E(J+1))GO TO 7 00001886
  6 CONTINUE                00001887
  7 JA=J                  00001888
      IF(JA.GE.JB)JA=JB-1    00001889
      DL1(I)=SLL(I)*(COS(ALP)+F*((P(I+1,JA+1)-P1(JA+1))/(2.*DS)+ 00001890
      1*(YL(I)-E(JA+1))*(P(I+1,JA+1)-P(I+1,JA)-P1(JA+1)+P1(JA))/(2.*DS*DE) 00001891
      2))-SIN(ALP)-GB*((-4.*P(I,JA)+P(I,JA-1))/(2.*DE)+(YL(I)-E(JA+1))* 00001892
      3*(-2.*P(I,JA)+P(I,JA-1))/(DE**2)) 00001893
      DL1(I)=DL1(I)/(1.5*GB/DE+(YL(I)-E(JA+1))/(DE**2)*GB) 00001894
      DL2(I)=2.*DL1(I)-P(I,JA) 00001895
      P(I,JA+1)=DL1(I) 00001896
      P(I,JA+2)=DL2(I) 00001897
      CALL SOLVE(2,JA) 00001898
      RS(2)=RS(2)-SUB(1)*P(I,1) 00001899
      RS(JA)=RS(JA)-SUP(JA)*P(I,JA+1) 00001900
      CALL TRID(2,JA) 00001901
      DO 9 J=2,JA 00001902
      DP=ABS(P(I,J)-RS(J)) 00001903
      IF(DP.GT.DPM)ICON=I 00001904
      IF(DP.GT.DPM)JCON=J 00001905
      IF(DP.GT.DPM)DPM=DP 00001906
      P2(J)=P1(J) 00001907
      P1(J)=P(I,J) 00001908
  9 P(I,J)=RS(J) 00001909
      JAM1=JA+1 00001910

```

```

    DO 10 J=JAM1,JB          00001911
    P2(J)=P1(J)              00001912
10 P1(J)=P(I,J)            00001913
    P2(I)=P1(I)              00001914
    P1(I)=P(I,1)              00001915
    JB1=JA                  00001916
    DL1(I)=SLL(I)*(COS(ALP)+F*((P(I+1,JA+1)-P(I-1,JA+1))/(2.*DS)+  

1*(YL(I)-E(JA+1))*(P(I+1,JA+1)-P(I+1,JA)-P(I-1,JA+1)+P(I-1,JA))/(2.*  

2*DS*DE))-SIN(ALP)-GB*((-4.*P(I,JA)+P(I,JA-1))/(2.*DE)+(YL(I)-E(  

3JA+1))*(-2.*P(I,JA)+P(I,JA-1))/(DE**2))
    DL1(I)=DL1(I)/(1.5*GB/DE+(YL(I)-E(JA+1))/(DE**2)*GB)        00001917
    DL2(I)=2.*DL1(I)-P(I,JA)          00001918
    P(I,JA+1)=DL1(I)          00001919
    P(I,JA+2)=DL2(I)          00001920
2500 CONTINUE                00001921
2501 CONTINUE                00001922
    DO 11 I=ILE,I1          00001923
    P(I,JB+1)=TEMP(I)          00001924
    TEMP1=PB(I)              00001925
    PB(I)=P(I,JB)              00001926
11 P(I,JB)=TEMP1          00001927
    TEMP1=TEMP4              00001928
    TEMP4=P1(JB)              00001929
    P1(JB)=TEMP1              00001930
    TEMP1=TEMP3              00001931
    TEMP3=P2(JB)              00001932
    P2(JB)=TEMP1              00001933
    PB(ILE1)=P(ILE1,JB)          00001934
    PB(ILE-2)=P(ILE-2,JB)          00001935
    RETURN                    00001936
    END                      00001937

```

```

SUBROUTINE WAKE                               00001942
C                                              00001943
C ***** SOLVES FLOW BEHIND THE AIRFOIL ***** 00001944
C                                              00001945
C
      REAL M
      COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)00001947
      1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),          00001948
      1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(9900001949
      2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),          00001950
      3A1,A2,A12,ALP,CIR,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,00001951
      4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP100001952
      5,Q,QQ,UUJB1,PI,P12,A22,A11,X4,S4                      00001953
      COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1, 00001954
      1JMAX,JCON,JMAX1,NSSP,IW                           00001955
      COMMON/ADAM/VJB,VJB1,VJBP1                         00001956
C RELAXES FLOW IN WAKE DIRECTLY               00001957
      IW=1                                         00001958
      DO 4000 I=ITE1,IMAX1                         00001959
      F=FF(I)
      FP12=FFP12(I)                                00001960
      FM12=FFM12(I)                                00001961
      FM32=FFM32(I)                                00001962
      CALL SOLVE(2,JMAX1)                          00001963
      RS(2)=RS(2)-SUB(1)*P(I,1)                    00001964
      RS(JMAX1)=RS(JMAX1)-SUP(JMAX1)*P(I,JMAX)    00001965
      IF(QQJBP1.LE.AAJBP1)GO TO 1                  00001966
      IF(VJBP1.LT.0.)GO TO 1                       00001967
      G=A11/(1.+TAN(PI2*E(JB+1))**2)            00001968
      GM32=A11/(1.+TAN(PI2*(E(JB+1)-1.5*DE))**2) 00001969
      RS(JB+1)=RS(JB+1)-(1.-QQJBP1/AAJBP1)*(VVJBP1/QQJBP1*G*GM32*CIR/ 00001970
      1(DE**2))                                     00001971
1 IF(QQJB.LE.AAJB)GO TO 2                     00001972
      IF(VJB.LT.0.0)GO TO 3                        00001973
      G=A11/(1.+TAN(PI2*E(JB))**2)              00001974
      GM12=A11/(1.+TAN(PI2*(E(JB)-0.5*DE))**2) 00001975
      RS(JB)=RS(JB)+(1.-QQJB/AAJB)*(VVJB/QQJB*G*GM12*CIR/(DE**2)) 00001976
      1-UUJB/QQJB*G*GM12*CIR/(DE**2)           00001977
      GO TO 4                                     00001978
3 G=A11/(1.+TAN(PI2*E(JB))**2)              00001979
      GM12=A11/(1.+TAN(PI2*(E(JB)-0.5*DE))**2) 00001980

```

```

    RS(JB)=RS(JB)-UUJB/QQJB*G*GM12*CIR/(DE**2)          00001982
    GO TO 4                                              00001983
2 G=A11/(1.+TAN(PI2*E(JB))**2)                      00001984
    C=(1.-VVJB/AAJB)*G                                     00001985
    GM12=A11/(1.+TAN(PI2*(E(JB)-0.5*DE))**2)           00001986
    RS(JB)=RS(JB)-C*GM12*CIR/(DE**2)                   00001987
4 IF(QQJB1.GT.AAJB1)GO TO 5                         00001988
    G=A11/(1.+TAN(PI2*E(JB-1))**2)                     00001989
    C=(1.-VVJB1/AAJB1)*G                                00001990
    GP12=A11/(1.+TAN(PI2*(E(JB-1)+0.5*DE))**2)        00001991
    RS(JB-1)=RS(JB-1)+C*GP12*CIR/(DE**2)              00001992
    GO TO 6                                              00001993
5 IF(VJB1.LT.0.0)GO TO 7                           00001994
    G=A11/(1.+TAN(PI2*E(JB-1))**2)                     00001995
    GP12=A11/(1.+TAN(PI2*(E(JB-1)+0.5*DE))**2)       00001996
    RS(JB-1)=RS(JB-1)+UUJB1/QQJB1*G*GP12*CIR/(DE**2)  00001997
    GO TO 6                                              00001998
7 G=A11/(1.+TAN(PI2*E(JB-1))**2)                   00001999
    GP12=A11/(1.+TAN(PI2*(E(JB-1)+0.5*DE))**2)       00002000
    RS(JB-1)=RS(JB-1)-(1.-QQJB1/AAJB1)*VVJB1/QQJB1*G*GP12*CIR/(DE**2) 00002001
    +UUJB1/QQJB1*G*GP12*CIR/(DE**2)                  00002002
6 CALL TRID(2,JMAX1)                               00002003
    DO 8 J=2,JMAX1                                    00002004
    DP=ABS(P(I,J)-RS(J))                            00002005
    IF(DP.LT.DPM)GO TO 9                          00002006
    ICQN=I                                         00002007
    JCQN=J                                         00002008
    DPM=DP                                         00002009
9 P2(J)=P1(J)                                       00002010
    P1(J)=P(I,J)                                    00002011
8 P(I,J)=RS(J)                                     00002012
    P2(1)=P1(1)                                    00002013
    P2(JMAX)=P1(JMAX)                            00002014
    P1(1)=P(I,1)                                    00002015
    P1(JMAX)=P(I,JMAX)                           00002016
4000 CONTINUE                                     00002017
    DO 10 I=ITE1,IMAX1                            00002018
    PE(I)=P(I,JB)-CIR                           00002019
10 CONTINUE                                      00002020
    PE(ITE1)=P(ITE1,JB)                           00002021

```

IW=0	
RETURN	00002022
END	00002023
	00002024

SUBROUTINE SHAPE	00002025
C	00002026
C ***** COMPUTES SHAPE OF AIRFOIL IN INVERSE DESIGN CASE*****	00002027
C	00002028
REAL M	00002029
COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)	00002030
1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),	00002031
1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(9900002032	
2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),	00002033
3A1,A2,A12,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,	00002034
4W,X1,X2,VVJB,VVJB1,AAJB,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP1	00002035
5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4	00002036
COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1,	00002037
1JMAX,JCON,JMAX1,NSSP,IW	00002038
COMMON ATAMU/DELTA Y	00002039
DELTA Y=0.0	00002040
IF(INV.EQ.0)I1=ILE+2	00002041
IP1=I1	00002042
DO 1 I=IP1,ITE1	00002043
YOLD=YU(I)	00002044
JB2=JB-2	00002045
DO 3 J=JB2,JMAX1	00002046
IF(YU(I-1).GT.E(J).AND.YU(I-1).LE.E(J+1))GO TO 4	00002047
3 CONTINUE	00002048
4 JA=J+1	00002049
L=I-1	00002050
IF(JA.LE.JB)JA=JB+1	00002051
F=FF(L)	00002052
U=QI*(COS(ALP)+F*((P(I,JA-1)-P(I-2,JA-1))/(2.*DS)+(YU(L)-E(JA-1)	00002053
1)*(P(I,JA)-P(I,JA-1)-P(I-2,JA)+P(I-2,JA-1))/(2.*DS*DE)))	00002054
GB=A11*(1.+TAN(P12*YU(I-1))**2)	00002055
Y=QI*(SIN(ALP)+GB*(-3.*P(L,JA-1)+4.*P(L,JA)-P(L,JA+1))/(2.*	00002056

```

1DE)+(YU(L)-E(JA-1))*(P(L,JA-1)-2.*P(L,JA)+P(L,JA+1))/(DE**2))) 00002057
FY=(GB/F)*V/U 00002058
IF(I.EQ.I1)GO TO 14 00002059
SLU(I-1)=V/U 00002060
14 CONTINUE 00002061
FK1=DS*FY 00002062
YN=YU(L)+0.5*FK1 00002063
F=FFM12(I) 00002064
DO 20 J=JB2,JMAX1 00002065
IF(YN.GT.E(J).AND.YN.LE.E(J+1))GO TO 50 00002066
20 CONTINUE 00002067
50 JA=J+1 00002068
IF(JA.LE.JB)JA=JB+1 00002069
U=QI*(COS(ALP)+F*((P(I,JA-1)-P(I-1,JA-1))/DS+(YN-E(JA-1))* 00002070
1*(P(I,JA)-P(I,JA-1)-P(L,JA)+P(L,JA-1))/(DS*DE))) 00002071
GB=A11/(1.+TAN(PI2*YN)**2) 00002072
V=QI*(SIN(ALP)+GB*(-3.*((P(I,JA-1)+P(L,JA-1))+4.*((P(I,JA)+P(L, 00002073
1JA))-P(I,JA+1)-P(L,JA+1))/(4.*DE)+(YN-E(JA-1))*0.5*(P(I,JA-1) 00002074
2+P(I-1,JA-1)-2.*((P(I,JA)+P(L,JA))+P(I,JA+1)+P(L,JA+1))/(DE**2))) 00002075
FK2=GB/F*DS*V/U 00002076
YN=YU(L)+0.5*FK2 00002077
DO 21 J=JB2,JMAX1 00002078
IF(YN.GT.E(J).AND.YN.LE.E(J+1))GO TO 22 00002079
21 CONTINUE 00002080
22 JA=J+1 00002081
IF(JA.LE.JB)JA=JB+1 00002082
U=QI*(COS(ALP)+F*((P(I,JA-1)-P(I-1,JA-1))/DS+(YN-E(JA-1))* 00002083
1*(P(I,JA)-P(I,JA-1)-P(L,JA)+P(L,JA-1))/(DS*DE))) 00002084
GB=A11/(1.+TAN(PI2*YN)**2) 00002085
V=QI*(SIN(ALP)+GE*(-3.*((P(I,JA-1)+P(L,JA-1))+4.*((P(I,JA)+P(L, 00002086
1JA))-P(I,JA+1)-P(L,JA+1))/(4.*DE)+(YN-E(JA-1))*0.5*(P(I,JA-1) 00002087
2+P(I-1,JA-1)-2.*((P(I,JA)+P(L,JA))+P(I,JA+1)+P(L,JA+1))/(DE**2))) 00002088
FK3=GB/F*DS*V/U 00002089
YN=YU(L)+FK3 00002090
F=FF(I) 00002091
DO 2 J=JB2,JMAX1 00002092
IF(YN.GT.E(J).AND.YN.LE.E(J+1))GO TO 5 00002093
2 CONTINUE 00002094
5 JA=J+1 00002095
IF(JA.LE.JB)JA=JB+1 00002096

```

```

U=QI*(COS(ALP)+F*((P(I+1,JA-1)-P(I-1,JA-1))/(2.*DS)+(YN-E(JA-1))* 00002097
1(P(I+1,JA)-P(I+1,JA-1)-P(I-1,JA)+P(I-1,JA-1))/(2.*DS*DE))) 00002098
GB=A11/(1.+TAN(PI2*YN)**2) 00002099
V=QI*(SIN(ALP)+GB*((-3.*P(I,JA-1)+4.*P(I,JA)-P(I,JA+1))/(2.*DE) 00002100
1+(YN-E(JA-1))*(P(I,JA-1)-2.*P(I,JA)+P(I,JA+1))/(DE**2))) 00002101
FK4=GB/F*DS*V/U 00002102
YU(I)=YU(L)+(FK1+2.*FK2+2.*FK3+FK4)/6. 00002103
IF(I.GT.ITE)GO TO 1 00002104
CHANGE=ABS(YU(I)-YOLD) 00002105
IF(CHANGE.GT.DELTAY)DELTAY=CHANGE 00002106
1 CONTINUE 00002107
DO 6 I=ILE1,IMAX 00002108
TEMP1=P(I,JB) 00002109
P(I,JB)=PB(I) 00002110
6 PB(I)=TEMP1 00002111
DO 7 I=IP1,ITE1 00002112
YOLD=YL(I) 00002113
JB2=JB+2 00002114
DO 8 JJ=1,JB 00002115
J=JB2-JJ 00002116
IF(YL(I-1).GE.E(J).AND.YL(I-1).LT.E(J+1))GO TO 9 00002117
8 CONTINUE 00002118
9 JA=J 00002119
IF(JA.GE.JB)JA=JB-1 00002120
L=I-1 00002121
F=FF(L) 00002122
U=QI*(COS(ALP)+F*((P(I,JA+1)-P(I-2,JA+1))/(2.*DS)+(YL(L)-E(JA+1))*00002123
1(P(I,JA+1)-P(I,JA)-P(I-2,JA+1)+P(I-2,JA))/(2.*DS*DE))) 00002124
GB=A11/(1.+TAN(PI2*YL(L)**2) 00002125
V=QI*(SIN(ALP)+GB*((3.*P(L,JA+1)-4.*P(L,JA)+P(L,JA-1))/(2.*DE)+ 00002126
1(YL(L)-E(JA+1))*(P(L,JA+1)-2.*P(L,JA)+P(L,JA-1))/(DE**2))) 00002127
FY=GB/F*V/U 00002128
IF(I.EQ.I1)GO TO 15 00002129
SLL(I-1)=V/U 00002130
15 CONTINUE 00002131
FK1=DS*FY 00002132
YN=YL(L)+0.5*FK1 00002133
F=FFM12(I) 00002134
DO 25 JJ=1,JB 00002135
J=JB2-JJ 00002136

```

```

    IF(YN.GE.E(J).AND.YN.LT.E(J+1))GO TO 26          00002131
25 CONTINUE                                         00002138
26 JA=J                                           00002139
    IF(JA.GE.JB)JA=JB-1                           00002140
    U=QI*(COS(ALP)+F*((P(I,JA+1)-P(L,JA+1))/DS+(YN-E(JA+1))*(P(I,JA+1)00002141
1-P(I,JA)-P(L,JA+1)+P(L,JA))/(DS*DE)))           00002142
    GB=A11/(1.+TAN(PI2*YN)**2)                     00002143
    V=QI*(SIN(ALP)+GB*((3.*(P(I,JA+1)+P(L,JA+1))-4.*(P(I,JA)+P(L,JA)) 00002144
1+P(I,JA-1)+P(L,JA-1))/(4.*DE)+(YN-E(JA+1))*(P(I,JA+1)+P(L,JA+1) 00002145
2-2.*((P(I,JA)+P(L,JA))+P(I,JA-1)+P(L,JA-1))*0.5/(DE**2)))        00002146
    FK2=GB/F*DS*V/U                               00002147
    YN=YL(L)+0.5*FK2                            00002148
    DO 27 JJ=1,JB                                00002149
    J=JB2-JJ                                     00002150
    IF(YN.GE.E(J).AND.YN.LT.E(J+1))GO TO 28          00002151
27 CONTINUE                                         00002152
28 JA=J                                           00002153
    IF(JA.GE.JB)JA=JB-1                           00002154
    U=QI*(COS(ALP)+F*((P(I,JA+1)-P(L,JA+1))/DS+(YN-E(JA+1))*(P(I,JA+1)00002155
1-P(I,JA)-P(L,JA+1)+P(L,JA))/(DS*DE)))           00002156
    GB=A11/(1.+TAN(PI2*YN)**2)                     00002157
    V=QI*(SIN(ALP)+GB*((3.*(P(I,JA+1)+P(L,JA+1))-4.*(P(I,JA)+P(L,JA)) 00002158
1+P(I,JA-1)+P(L,JA-1))/(4.*DE)+(YN-E(JA+1))*(P(I,JA+1)+P(L,JA+1) 00002159
2-2.*((P(I,JA)+P(L,JA))+P(I,JA-1)+P(L,JA-1))*0.5/(DE**2)))        00002160
    FK3=GB/F*DS*V/U                               00002161
    YN=YL(L)+FK3                                 00002162
    F=FF(I)                                       00002163
    DO 10 JJ=1,JB                                00002164
    J=JB2-JJ                                     00002165
    IF(YN.GE.E(J).AND.YN.LT.E(J+1))GO TO 11          00002166
10 CONTINUE                                         00002167
11 JA=J                                           00002168
    IF(JA.GE.JB)JA=JB-1                           00002169
    U=QI*(COS(ALP)+F*((P(I+1,JA+1)-P(I-1,JA+1))/(2.*DS)+(YN-E(JA+1))* 00002170
1(P(I+1,JA+1)-P(I+1,JA)-P(I-1,JA+1)+P(I-1,JA))/(2.*DS*DE)))         00002171
    GB=A11/(1.+TAN(PI2*YN)**2)                     00002172
    V=QI*(SIN(ALP)+GB*((3.*P(I,JA+1)-4.*P(I,JA)+P(I,JA-1))/(2.*DE)+ 00002173
1(YN-E(JA+1))*(P(I,JA+1)-2.*P(I,JA)+P(I,JA-1))/(DE**2)))        00002174
    FK4=GB/F*DS*V/U                               00002175
    YL(I)=YL(L)+(FK1+2.*FK2+2.*FK3+FK4)/6.        ,00002176

```

```

    IF(I.GT.ITE)GO TO 7          00002177
    CHANGE=ABS(YL(I)-YOLD)        00002178
    IF(CHANGE.GT.DELTAY)DELTAY=CHANGE 00002179
7 CONTINUE                      00002180
    DO 12 I=ILE1,IMAX           00002181
    TEMP1=P(I,JB)                00002182
    P(I,JB)=PB(I)                00002183
12 PB(I)=TEMP1                  00002184
    RETURN                       00002185
    END                          00002186
    SUBROUTINE TRID(IL,IH)       00002187
C                                     00002188
C ***** TRIDIAGONAL EQUATION SOLVER *****
C                                     00002189
C                                     00002190
    REAL M                      00002191
    COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)00002192
    1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),          00002193
    1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SUB(99),TEMP(99)00002194
    2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),          00002195
    3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,00002196
    4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBPI,QQJBPI,AAJBP100002197
    5,Q,QQ,UUJB1,PI,PI2,A22,A11,X4,S4                      00002198
    COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1, 00002199
    1JMAX,JCON,JMAX1,NSSP,IW                      00002200
    N=IH                      00002201
    NN=N-1                      00002202
    SUP(IL)=SUP(IL)/D(IL)        00002203
    RS(IL)=RS(IL)/D(IL)        00002204
    IDUM=IL+1                  00002205
    DO 10 L=IDUM,N              00002206
    II=L-1                      00002207
    D(L)=D(L)-SUP(II)*SUB(II)  00002208
    IF(L.EQ.N)GO TO 10          00002209
    SUP(L)=SUP(L)/D(L)          00002210
10 RS(L)=(RS(L)-SUB(II)*RS(II))/D(L)  00002211
    DO 20 K=IL,NN              00002212
    L=N-K+IL-1                  00002213
20 RS(L)=RS(L)-SUP(L)*RS(L+1)  00002214
    RETURN                      00002215
    END                        00002216

```

```

SUBROUTINE HALVE                                00002217
C                                                00002218
C ***** THIS SUBROUTINE HALVES THE GRID SPACING ETC. *****
C                                                00002219
C                                                00002220
REAL M                                         00002221
COMMON CPU(99),CPL(99),E(99),DU1(99),DU2(99),DL1(99),DL2(99),D(99)00002222
1,FF(99),FFP12(99),FFM12(99),FFM1(99),FFM32(99),                           00002223
1P1(99),P2(99),PB(99),P(99,99),RS(99),S(99),SUP(99),SU8(99),TEMP(9900002224
2),X(99),Y(99),YU(99),YL(99),SLU(99),SLL(99),                           00002225
3A1,A2,AI2,ALP,CIR,EPS,EPSS,DE,DS,DP,DPM,F,FP12,FM12,FM32,M,QI,QI2,00002226
4W,X1,X2,VVJB,VVJB1,AAJB1,AAJB,QQJB,QQJB1,UUJB,VVJBP1,QQJBP1,AAJBP100002227
5,QQ,QQ,UUJB1,PI,PI2,A22,A11,X4,S4                                         00002228
COMMON I,ITE,ITE1,ILE,ILE1,I1,I11,ICON,IMAX,IMAX1,INV,JB,JA1,JB1, 00002229
1JMAX,JCON,JMAX1,NSSP,IW                                         00002230
IMAX=2*IMAX-1                                         00002231
JMAX=2*JMAX-1                                         00002232
IMAX1=IMAX-1                                         00002233
JMAX1=JMAX-1                                         00002234
J=JMAX1/2+1                                         00002235
JJ=JMAX                                         00002236
1 I=IMAX1/2+1                                         00002237
II=IMAX                                         00002238
2 P(II,JJ)=P(I,J)                                         00002239
I=I-1                                         00002240
II=II-2                                         00002241
IF(I.GT.0)GO TO 2                                         00002242
J=J-1                                         00002243
JJ=JJ-2                                         00002244
IF(J.GT.0)GO TO 1                                         00002245
DO 3 J=1,JMAX,2                                         00002246
DO 3 I=2,IMAX1,2                                         00002247
3 P(I,J)=0.5*(P(I+1,J)+P(I-1,J)) 00002248
DO 4 I=1,IMAX                                         00002249
DO 4 J=2,JMAX1,2                                         00002250
4 P(I,J)=0.5*(P(I,J+1)+P(I,J-1)) 00002251
I=IMAX1/2+1                                         00002252
II=IMAX                                         00002253
5 PB(II)=PB(I)                                         00002254
I=I-1                                         00002255
II=II-2                                         00002256

```

```

IF(I.GT.0)GO TO 5
DO 6 I=2,IMAX1,2
6 PB(I)=0.5*(PB(I+1)+PB(I-1))
RETURN
END

SUBROUTINE PLOT(NG,A,N,M,NL,NS)
***** THIS CREATES A PLOT OF RESULTS ON THE STANDARD PRINT OUT **00002264
DIMENSION OUT(101),YPR(11),ANG(9),A(500)
1 FORMAT(1H1,60X,7H CHART ,I3,//)
2 FORMAT(1H ,F11.4,5X,10I1)
3 FORMAT(1H )
7 FORMAT(1H ,16X,10I1.
1   .   .   .   .   .   .   .   .   .   .)
8 FORMAT(1H0,9X,11F10.3)
DATA BLANK/1H /,ANG/1HU,1HL,1HT,1HB,1H5,1H6,1H7,1H8,1H9/
NLL=NL
IF(NS) 16, 16, 10
0 DO 15 I=1,N
DO 14 J=I,N
IF(A(I)-A(J)) 14, 14, 11
1 L=I-N
LL=J-N
DO 12 K=1,M
L=L+N
LL=LL+N
F=A(L)
A(L)=A(LL)
2 A(LL)=F
4 CONTINUE
5 CCONTINUE
6 IF(NLL) 20, 18, 20
8 NLL=50
0 WRITE(6,1)NO
XSCAL=(A(N)-A(1))/(FLOAT(NLL-1))
M1=N+1
00002257
00002258
00002259
00002260
00002261
00002262
00002263
00002265
00002266
00002267
00002268
00002269
00002270
00002271
00002272
00002273
00002274
00002275
00002276
00002277
00002278
00002279
00002280
00002281
00002282
00002283
00002284
00002285
00002286
00002287
00002288
00002289
00002290
00002291
00002292
00002293

```

```

YMIN=A(M1)          00002294
YMAX=YMIN          00002295
M2=M*N             00002296
DO 40 J=M1,M2      00002297
IF(A(J)-YMIN) 28,26,26
26 IF(A(J)-YMAX) 40,40,30
28 YMIN=A(J)
GO TO 40
30 YMAX=A(J)
40 CONTINUE
YSCAL=((YMAX-YMIN)/100.0
XB=A(1)
L=1
MY=M-1
I=1
45 F=I-1
XPR=XB+F*XSCAL
IF(A(L)-XPR) 50,50,70
50 DO 55 IX=1,101
55 OUT(IX)=BLANK
DO 60 J=1,MY
LL=L+J*N
JP=((A(LL)-YMIN)/YSCAL)+1.0
OUT(JP)=ANG(J)
60 CONTINUE
WRITE(6,2)XPR,(OUT(IZ),IZ=1,101)
L=L+1
GOTO80
70 WRITE(6,3)
80 I=I+1
IF(I-NLL) 45,84,86
84 XPR=A(N)
GO TO 50
86 WRITE(6,7)
YPR(1)=YMIN
DO 90 KN=1,9
90 YPR(KN+1)=YPR(KN)+YSCAL*10.0
YPR(11)=YMAX

```

```

      WRITE(6,8)(YPR(IP),IP=1,11)
      RETURN
      END

      SUBROUTINE ARC(XI,YI,XO,YO,SI,SO,XP,YP,D1Y,D2Y,D3Y,DERIX,DERFX,
      1DERIY,DERFY,NI,NO,INT)
      ****      DETERMINES THE ARC LENGTH OF THE AIRFOIL POINTS ****

      DIMENSION XI(1),YI(1),XO(1),YO(1),SI(1),SO(1),XP(1),YP(1),D1Y(1),
      1D2Y(1),D3Y(1)
      SI - INPUT CHORD LENGTH          SO - OUTPUT CHORD LENGTH
      COMPUTE ARC LENGTH SI USING CIRCULAR ARC SEGMENTS
      INT=1      SPLINE XI AND YI VS SI
      EPSI=1.E-10
      NI=NI-1
      SI(1)=0.
      H1=0.
      DX1=XI(2)-XI(1)
      DY1=YI(2)-YI(1)
      C1=SQRT(DX1**2+DY1**2)
      SI(2)=C1
      IF(NI.EQ.2)RETURN
      DO 1 I=2,NI
      DX1=XI(I)-XI(I-1)
      DY1=YI(I)-YI(I-1)
      DX2=XI(I+1)-XI(I)
      DY2=YI(I+1)-YI(I)
      DX=XI(I+1)-XI(I-1)
      DY=YI(I+1)-YI(I-1)
      C2=SQRT(DX2**2+DY2**2)
      C=SQRT(DX**2+DY**2)
      A=(DY1*DX-DY*DX1)/2.
      H=4.*A/(C*C1*C2)
      HAV=(H1+H)/2.
      DS=C1*(1.+(C1/2.*HAV)**2/6.)
      SI(I)=SI(I-1)+DS
      C1=C2
      H1=H
1     CONTINUE

```

```

DS=C1*(1.+(C1/2.*H)**2/6.)
SI(NI)=SI(NI-1)+DS
IF(INT.NE.1)RETURN
2 CONTINUE
C SPLINE XI AS A FUNCTION OF SI
CALL SPLINE(SI,XI,SO,XG,XP,D1Y,D2Y,D3Y,1,3,DERIX,DERFX,NI,NO,0) 00002371
00002372
00002373
00002374
00002375
00002376
00002377
00002378
00002379
00002380
00002381

3 CONTINUE
C SPLINE YI AS A FUNCTION OF SI
CALL SPLINE(SI,YI,SO,YO,YP,D1Y,D2Y,D3Y,1,3,DERIY,DERFY,NI,NO,1)
RETURN
END

SUBROUTINE SPLINE(XIN,YIN,XOUT,YOUT,DYDX,D1Y,D2Y,D3Y,NDERI,NDERF, 00002382
1DERIVI,DERIVF,NIN,NOUT,INTERP) 00002383
C E B KLUNKER JANUARY 1973 00002384
C COMPUTE A CUBIC SPLINE THROUGH THE SET OF POINTS XIN(I),YIN(I) 00002385
C XIN MUST BE MONOTONIC 00002386
C XIN,YIN INPUT INDEPENDENT AND DEPENDENT VARIABLES 00002387
C XOUT,YOUT OUTPUT INDEPENDENT AND DEPENDENT VARIABLES 00002388
C D1Y,D2Y,D3Y 1ST, 2ND, AND 3RD DERIVATIVE AT SPLINE POINTS XIN 00002389
C DYDX DERIVATIVE AT XOUT 00002390
C NIN ,NOUT NUMBER OF INPUT AND OUTPUT VALUES 00002391
C NDERI ORDER OF DERIVATIVE AT INITIAL SPLINE POINT (1,2,OR 3) 00002392
C NDERF ORDER OF DERIVATIVE AT FINAL SPLINE POINT (1,2,OR 3) 00002393
C DERIVI VALUE OF DERIVATIVE AT INITIAL SPLINE POINT 00002394
C INTERP NE 1 INTERPOLATE FOR GIVEN VALUES YOUT 00002395
C NTIMES NE 1 SPLINE COEFFICIENTS ARE NOT RECOMPUTED 00002396
DIMENSION XIN(1),YIN(1),XOUT(1),YOUT(1),DYDX(1),D1Y(1),D2Y(1), 00002397
1D3Y(1) 00002398
EPSI1=-1.E-10 00002399
EPSI2=-EPSI1 00002400
NIM1=NIN-1 00002401
DX=XIN(2)-XIN(1) 00002402
I=2 00002403
IF(DX.EQ.0.)GO TO 35 00002404
DF=(YIN(2)-YIN(1))/DX 00002405
IF(NDERI-2)1,2,3 00002406
1 C=.5 00002407

```

```

        F=3.*(DF-DERIVI)/DX          00002408
      GO TO 4                      00002409
2 C=0.                         00002410
      F=DERIVI                     00002411
      GO TO 4                      00002412
3 C=-1.                         00002413
      F=-DX*DERIVI                 00002414
C   FORWARD LOOP OF TRIDIAGONAL MATRIX COMPUTATION 00002415
4 D1Y(1)=-C                     00002416
      D2Y(1)=F                     00002417
      DO 5 I=2,NIM1                00002418
      DX1=XIN(I+1)-XIN(I)          00002419
      IF(DX1.EQ.0.)GO TO 36        00002420
      DF1=(YIN(I+1)-YIN(I))/DX1   00002421
      B=2.*(DX+DX1)                00002422
      F=6.*(DF1-DF)                00002423
      DENOM=B+DX*D1Y(I-1)          00002424
      D2Y(I)=(F-DX*D2Y(I-1))/DENOM 00002425
      D1Y(I)=-DX1/DENOM           00002426
      DX=DX1                       00002427
      DF=DF1                       00002428
5 CONTINUE                      00002429
      I=NIN                        00002430
      IF(NDERF>2)6,7,8             00002431
6 A=.5                          00002432
      F=-3.*(DF1-DERIVF)/DX1     00002433
      GO TO 9                      00002434
7 A=0.                          00002435
      F=DERIVF                     00002436
      GO TO 9                      00002437
8 A=-1.                         00002438
      F=DX1*DERIVF                 00002439
9 DENOM=1.+A*D1Y(I-1)            00002440
      D2Y(I)=(F-A*D2Y(I-1))/DENOM 00002441
      D1Y(I)=0.                     00002442
C   BACK SUBSTITUTION OF TRIDIAGONAL MATRIX COMPUTATION 00002443
      K=NIN                        00002444
      DO 11 I=1,NIM1                00002445
      K=K-1                         00002446
      D2Y(K)=D2Y(K)+D1Y(K)*D2Y(K+1) 00002447

```

```

10 DX1=XIN(K+1)-XIN(K) 0000244
  DF1=(YIN(K+1)-YIN(K))/DX1 00002449
  D1Y(K+1)=DF1+DX1/6.*D2Y(K)+2.*D2Y(K+1)) 00002450
  D3Y(K+1)=(D2Y(K+1)-D2Y(K))/DX1 00002451
11 CONTINUE 00002452
  D1Y(1)=DF1-DX1/6.*(2.*D2Y(1)+D2Y(2)) 00002453
  D3Y(1)=D3Y(2) 00002454
  IF(INTERP.NE.1)GO TO 16 00002455
C     INTERPOLATE FOR GIVEN VALUES OF XOUT 00002456
  DO 15 J=1,NOUT 00002457
  DO 12 I=1,NIN 00002458
  DX=XIN(I)-XOUT(J) 00002459
  IF(DX.GE.EPSI1.AND.DX.LE.EPSI2)GO TO 13 00002460
  IF(DX.GE.EPSI2)GO TO 14 00002461
12 CONTINUE 00002462
  GO TO 37 00002463
13 YOUT(J)=YIN(I) 00002464
  DYDX(J)=D1Y(I) 00002465
  GO TO 15 00002466
14 DX=XOUT(J)-XIN(I) 00002467
  YOUT(J)=YIN(I)+DX*(D1Y(I)+DX/2.*(D2Y(I)+DX/3.*D3Y(I))) 00002468
  DYDX(J)=D1Y(I)+DX*(D2Y(I)+DX/2.*D3Y(I)) 00002469
15 CONTINUE 00002470
  GO TO 23 00002471
C     INTERPOLATION FOR GIVEN VALUES OF YOUT 00002472
16 DO 22 J=1,NOUT 00002473
  DO 17 I=1,NIN 00002474
  DY=YIN(I)-YOUT(J) 00002475
  IF(DY.GE.EPSI1.AND.DY.LE.EPSI2)GO TO 18 00002476
  IF(DY.GE.EPSI2)GO TO 19 00002477
17 CONTINUE 00002478
  GO TO 38 00002479
18 YOUT(J)=YIN(I) 00002480
  XOUT(J)=XIN(I) 00002481
  DYDX(J)=D1Y(I) 00002482
  GO TO 22 00002483
19 DX=-DY*D1Y(I) 00002484
20 YO=YIN(I)+DX*(D1Y(I)+DX/2.*(D2Y(I)+DX/3.*D3Y(I))) 00002485
  DY=YO-YOUT(J) 00002486
  IF(DY.GE.EPSI1.AND.DY.LE.EPSI2)GO TO 21 00002487

```

```

`YP=D1Y(I)+DX*(D2Y(I)+DX/2.*D3Y(I))          00002488
DELX=-DY/YP          00002489
DX=DX+DELX          00002490
GO TO 20          00002491
21 XOUT(J)=XIN(I)+DX          00002492
    DYDX(J)=D1Y(I)+DX*(D2Y(I)+DX/2.*D3Y(I))  00002493
22 CCONTINUE          00002494
23 RETURN          00002495
35 PRINT 100          00002496
    PRINT 101,XIN(1),XIN(2)          00002497
    STOP          00002498
36 PRINT 100          00002499
    PRINT 102,I,XIN(I),XIN(I+1)          00002500
    STOP          00002501
37 PRINT 100          00002502
    PRINT 103,J,XOUT(J),XIN(NIN)          00002503
    STOP          00002504
38 PRINT 100          00002505
    PRINT 104,J,YOUT(J),YIN(NIN)          00002506
    STOP          00002507
C          00002508
100 FORMAT(/5X,*SUBROUTINE SPLINE*)          00002509
101 FORMAT(/5X,*ERROR IN INPUT   XIN(1)="E12.4.5X,"XIN(2)="E12.4/) 00002510
102 FORMAT(/5X,*ERROR IN INPUT   I="I5.5X,"XIN(I)="E12.4.5X,"XIN(I+1)=00002511
    "E12.4A)          00002512
103 FORMAT(/5X,*XOUT(J) IS OUT OF RANGE   J="I5.5X,"XOUT(J)="E12.4.5X,00002513
    "XIN(NIN)="E12.4/)          00002514
`104 FORMAT(/5X,*YOUT(J) IS OUT OF RANGE   J="I5.5X,"YOUT(J)="E12.4.5X,00002515
    "YIN(NIN)="E12.4/)          00002516
C          00002517
END          00002518

```

## APPENDIX D SAMPLE CASES

The input and output for two sample cases are presented on the following pages. Case 1 is a typical inverse design solution, and Case 2 is the analysis of a typical aft-cambered airfoil. Note that Case 2 only has two grid halvings. In actual usage, an analysis case would usually have one additional grid halving and would use a non-zero value for CDCORR in order to obtain accurate drag values (see Appendix B).

## Sample Case No. 1 - Inverse Airfoil Design

MACH 0.72 AIRFOIL DESIGN -- SAMPLE CASE 1  
 &INP M=0.72,X1=-0.38,X2=0.5,CONV=1.E-06,RN=20.95E06,XIBDLY=-0.34,XSFP=0.46,  
 &END  
 &INP INV=1,&END

0.50 10000.0

82

0.00000	0.00000	0.00012	0.00441	0.00082	0.00872	0.00210	0.01291
0.00397	0.01693	0.00647	0.02072	0.00966	0.02426	0.01361	0.02759
0.01833	0.03076	0.02379	0.03377	0.02999	0.03665	0.03692	0.03940
0.04457	0.04201	0.05293	0.04449	0.06199	0.04685	0.07174	0.04909
0.08217	0.05122	0.09327	0.05325	0.10499	0.05519	0.11733	0.05703
0.13025	0.05878	0.14374	0.06044	0.15775	0.06200	0.17228	0.06347
0.18729	0.06485	0.20276	0.06612	0.21866	0.06729	0.23495	0.06836
0.25163	0.06932	0.26865	0.07018	0.28598	0.07091	0.30361	0.07154
0.32150	0.07204	0.33961	0.07242	0.35793	0.07268	0.37642	0.07280
0.39505	0.07279	0.41380	0.07264	0.43262	0.07234	0.45150	0.07188
0.47042	0.07126	0.48934	0.07046	0.50825	0.06947	0.52714	0.06829
0.54599	0.06692	0.56481	0.06536	0.58359	0.06361	0.60232	0.06171
0.62098	0.05966	0.63956	0.05751	0.65802	0.05528	0.67634	0.05298
0.69447	0.05065	0.71239	0.04830	0.73005	0.04595	0.74742	0.04360
0.76446	0.04128	0.78113	0.03898	0.79740	0.03673	0.81323	0.03451
0.82860	0.03235	0.84346	0.03023	0.85780	0.02818	0.87158	0.02619
0.88478	0.02426	0.89736	0.02240	0.90932	0.02062	0.92061	0.01891
0.93123	0.01728	0.94116	0.01574	0.95036	0.01428	0.95882	0.01291
0.96654	0.01163	0.97349	0.01045	0.97967	0.00938	0.98504	0.00843
0.98960	0.00759	0.99334	0.00687	0.99624	0.00626	0.99832	0.00578
0.99958	0.00546	1.00000	0.00534				

0.0 1.0

0.0 0.0

80

0.00000	0.00000	0.00046	-0.00448	0.00150	-0.00901	0.00310	-0.01357
0.00524	-0.01812	0.00794	-0.02263	0.01119	-0.02704	0.01502	-0.03135
0.01943	-0.03553	0.02446	-0.03957	0.03009	-0.04347	0.03634	-0.04722
0.04321	-0.05082	0.05068	-0.05427	0.05876	-0.05756	0.06743	-0.06068
0.07669	-0.06364	0.08653	-0.06642	0.09694	-0.06902	0.10792	-0.07145
0.11945	-0.07369	0.13151	-0.07576	0.14410	-0.07765	0.15719	-0.07936
0.17077	-0.08090	0.18481	-0.08227	0.19929	-0.08346	0.21418	-0.08448
0.22946	-0.08532	0.24509	-0.08599	0.26106	-0.08647	0.27733	-0.08677

0.29387	-0.08688	0.31065	-0.08678	0.32765	-0.08647	0.34483	-0.08594
0.36217	-0.08517	0.37965	-0.08414	0.39723	-0.08283	0.41492	-0.08121
0.43270	-0.07926	0.45059	-0.07694	0.46861	-0.07426	0.48680	-0.07121
0.50519	-0.06782	0.52379	-0.06413	0.54262	-0.06019	0.56167	-0.05606
0.58093	-0.05179	0.60037	-0.04745	0.61997	-0.04308	0.63968	-0.03874
0.65947	-0.03448	0.67928	-0.03035	0.69906	-0.02638	0.71876	-0.02262
0.73831	-0.01911	0.75767	-0.01588	0.77676	-0.01296	0.79552	-0.01039
0.81386	-0.00819	0.83172	-0.00637	0.84902	-0.00495	0.86566	-0.00392
0.88158	-0.00327	0.89669	-0.00299	0.91093	-0.00303	0.92423	-0.00338
0.93653	-0.00398	0.94780	-0.00479	0.95799	-0.00575	0.96708	-0.00682
0.97504	-0.00794	0.98187	-0.00905	0.98758	-0.01010	0.99218	-0.01105
0.99566	-0.01185	0.99809	-0.01243	0.99952	-0.01278	1.00000	-0.01290
0.0	-1.0	0.0	0.0				
-0.38	.500						
-0.91200	-1.00600	-0.94000	-1.01500	-1.02000	-1.04900	-1.08100	-0.58300
-0.55700	-0.36000	-0.27500	-0.21500	-0.15400	-0.10000	-0.04100	
-0.69000	-0.68600	-0.63000	-0.62000	-0.61100	-0.56100	-0.30500	0.04000
0.21800	0.31500	0.40700	0.48100	0.53100	0.54700	0.57000	
-0.38	.500						
-1.10700	-1.10100	-1.09800	-1.06700	-1.07600	-1.07800	-1.06900	-1.06100
-1.06700	-1.05600	-1.05500	-1.01700	-0.91700	-0.80800	-0.70100	-0.60800
-0.52400	-0.43800	-0.37900	-0.32600	-0.28400	-0.24400	-0.20700	-0.16200
-0.14400	-0.11400	-0.08700	-0.05500	-0.00700			
-0.67300	-0.67700	-0.66500	-0.65700	-0.64900	-0.63700	-0.65000	-0.62800
-0.62100	-0.59000	-0.52800	-0.40800	-0.26700	-0.13900	0.00200	0.08900
0.16700	0.24100	0.29200	0.35700	0.39600	0.43600	0.47000	0.49800
0.51600	0.52800	0.53100	0.51000	0.53200			

MACH 0.72 AIRFCIL DESIGN -- SAMPLE CASE 1

X-Y GRID SYSTEM

```

2 -0.1410E 01   3 -0.4900E 00   4 -0.3706E 00   5 -0.2485E 00   6 -0.1247E 00   7  0.0
8  0.1247E 00   9  0.2485E 00   10 0.3706E 00  11 0.4900E 00  12 0.1410E 01
2 -0.4261E 00   3 -0.1420E 00   4 -0.2303E-07  5  0.1420E 00   6  0.4261E 00

MACH NO. IS 0.720 ANGLE OF ATTACK IS 0.0  DEGREES
DIRECT SOLUTION TC      0.50
CASE NUMBER 100

INVERSE DESIGN CASE
LFINP
M= 0.71499997 ,w= 1.6999998 ,xi= 0.50000000 ,x2= 10000.000 ,ALPE= 0.0 ,EPS= 0.0 ,EPSS=
0.99999998 ,x4= 0.48999995 ,S4= 2.0000000 ,CONV= 0.99999943E-06 ,AI= 0.24599999 ,A2= 0.14999998 ,AJ= 3.8649999
RN= 20950000.0 ,X1DLY= 0.34999990 ,CIR= 0.0 ,CDCHR= 0.0 ,RDEL= 0.25000000 ,KELFN= 0.12500000 ,
SP= 0.34999996E-02 ,XSEH= 0.45999998 ,RCB= 0.19499999 ,CPB= 0.34999998 ,XMON= 0.46999997 ,XLSEP= 0.50000000 ,XPC=
0.99999964E-01
EEND
EINP
IMAX= 13,JMAX= 7,IKASE= 100,INV= 1,MITER= 800,NHALF= 2,ITACT= 0,ISKP2=
0,ISKP3= 0,ISKP4= 0,ITERD= 0,IREAD= 0,LP= 1000,ITEUPC= 0,ITELWC=
0
EEND

AIRFOIL COORDINATES
X      YU      YL      UPMER SLOPE LOWER SLOPE
-0.49000 0.02458 -0.02552 0.03719 -1.31772
-0.37058 0.05667 -0.07542 0.12958 -0.16455
-0.24852 0.06531 -0.08621 0.05419 -0.03119
-0.12470 0.07280 -0.08942 0.00332 0.06268
0.0  0.06933 -0.06880 -0.05300 0.18773
0.12470 0.05924 -0.04293 -0.11419 0.21244
0.24852 0.04345 -0.01733 -0.13570 0.16651
0.37058 0.02634 -0.00368 -0.14519 0.04513
0.49000 0.00752 -0.01050 -0.18798 -0.20660
ITERATION 10 CIR = 0.10732 DPM = 0.00441951 AT 5 3 NSSP = 3 DELTAY OH DELSTAR = 0.0
ITERATION 20 CIR = 0.13470 DPM = 0.00171214 AT 12 3 NSSP = 4 DELTAY OH DELSTAR = 0.0
ITERATION 30 CIR = 0.15451 DPM = 0.00167090 AT 10 3 NSSP = 4 DELTAY OH DELSTAR = 0.0
ITERATION 40 CIR = 0.16997 DPM = 0.00133852 AT 10 3 NSSP = 4 DELTAY GR DELSTAR = 0.0
ITERATION 50 CIR = 0.18245 DPM = 0.00126487 AT 10 3 NSSP = 5 DELTAY OH DELSTAR = 0.0
CP BY BACKWARD DIFFERENCES
X          CPU          CPL
-0.490  -0.174  0.013
-0.371  -1.174  -1.043
-0.249  -0.633  -0.823
-0.125  -0.677  -0.789
0.0  0.798  -0.751
0.125  -0.567  -0.211
0.249  -0.474  -0.440
0.371  -0.067  -0.540
0.490  -0.342  -0.579
1.410  0.163  0.049
CP BY CENTRAL DIFFERENCES
X          CPU          CPL
-0.490  -0.174  0.013
-0.371  -1.174  -1.043
-0.249  -0.671  -0.909
-0.125  -0.785  -0.780
0.0  0.712  -0.442
0.125  -0.501  0.035
0.249  -0.264  0.337
0.371  -0.103  0.500
0.490  0.416  0.665
1.410  0.028  0.028
X      YU      VL      SLU      SLL
-0.49000 0.02458 -0.02552 0.03718 -1.31772
-0.37058 0.05667 -0.07542 0.12998 -0.16455
-0.24852 0.06531 -0.08621 0.05419 -0.03119
-0.12470 0.07280 -0.08942 0.00332 0.06268
0.0  0.06933 -0.06880 -0.05300 0.18773
0.12470 0.05924 -0.04293 -0.11419 0.2123
0.24852 0.04345 -0.01733 -0.13570 0.16651
0.37058 0.02634 -0.00368 -0.14519 0.04513
0.49000 0.00752 -0.01058 -0.18798 -0.20660
71 70 69 70 71
63 60 0 71 72
79 99 010J 62
83102 0102 6E
84 96 010G 65
78 86 0 97 67
73 72 0 89 63
69 63 0 81 75
67 60 0 76 75
51 45 0 60 6E
70 70 70 71 71

WAVE CD = -0.031180

```

CHART 100

-0.4920 U T L  
 -0.3700 U L T B  
 -0.2300 L U T B  
 -0.1100 . L T B  
 0.0100 U L T B  
 0.1300 U T B L B  
 0.2500 U T B L  
 0.3910 U T B L  
 0.4920 \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
 -1.174 -0.990 -0.000 -0.022 -0.038 -0.255 -0.071 0.113 0.297 0.481 0.665  
 PRESSURE COEFFICIENT  
 CPSTAR = -0.6993 CLCIR = 0.3650  
 CL = 0.5179 CD = -0.031180 CHLE = -0.1951 CDF = 0.0 CHMC4 = -0.1232

## MACH 0.72 AIRCRAFT DESIGN -- SAMPLE CASE I

### X-Y GRID SYSTEM

2	-0.3472E C1	3	-0.1410E 01	4	-0.6471E 00	5	-0.4900E 00	6	-0.4307E 00	7	-0.3706E 00
	0	9	-0.2485E 00	10	-0.1868E 00	11	-0.1247E 00	12	-0.06240E 01	13	0.0
14	0.6240E -1	15	0.1247E 00	16	0.1868E 00	17	0.2485E 00	18	0.3098E 00	19	0.3706E 00
20	0.417C0	21	3.4900E 00	22	0.6471E 00	23	0.1410E 01	24	0.3872E 01		
2	-0.9181E 00	3	-0.4261E 00	4	-0.2460E 00	5	-0.1420E 00	6	-0.0592E -01	7	-0.9213E -07
8	0.6562E -1	9	0.1420E 00	10	0.2460E 00	11	0.4261E 00	12	0.9181E 00		

MACH NO. IS 0.720 ANGLE OF ATTACK IS 0.0 DEGREES  
DIRECT SOLUTION TO -0.38  
CASE NUMBER 100

### INVERSE DESIGN CASE

```

*EFINP
  M= 0.71999997 ,X1= 1.0999998 ,X1I=-0.38000000 ,X2= 0.50000000 ,ALP= 0.0 ,EPSA= 0.0 ,EPSS=
  0.49999998 ,X4= 0.48999995 ,SA= 2.0000000 ,CUNV= 0.99939943E-06 ,AI= 0.24599999 ,A2= 0.14999998 ,A3=
  3.8699999 ,RNDL= 20950000 ,X1BDLY= 0.34999996 ,CIR= 0.1d249.27 ,CDCRK= 0.0 ,RDEL= 0.25000000 ,RDEFLN= 0.12500000 ,
  SP= 0.39999996E-02 ,XSEPI= 0.45999998 ,RCPB= 0.19999999 ,CPB= 0.39999998 ,XMON= 0.46999997 ,XLSEP= 0.50000000 ,XPC=
  0.99999994E-01
  END
  D1NP
  IMAX= 13 ,IKASE= 100 ,INV= 1 ,MITER= 400 ,NHALF= 2 ,ITACT= 0 ,ISKP2=
  0 ,ISKP3= 0 ,ISK3= 0 ,ITERP= 0 ,IHEAD= 0 ,LP= 1000 ,ITEUPC= 0 ,ITELWC=
  0
  END

```

X	Y_U	Y_L	UPHIL SLOPE	LAWN SLOPE
-0.44300	0.24548	-0.00156	0.91718	-1.31772
-0.44361	0.24548	-0.00132	0.22258	-0.33309
-0.37036	0.20567	-0.07542	0.15656	-0.16655
-0.30984	0.16513	-0.08274	0.08500	-0.08394
-0.24863	0.12331	-0.04621	0.02619	-0.03114
-0.18777	0.07143	-0.00473	0.02750	-0.01356
-0.12970	0.02740	-0.00442	0.32332	0.06228
-0.06943	0.07721	-0.07400	-0.02260	0.12527
0.0	0.00593	-0.00500	-0.05300	0.18773
0.06220	0.06557	-0.05550	-0.08885	0.21997
0.12476	0.05924	-0.062201	-0.11414	0.22123
0.18677	0.05112	-0.02683	-0.12870	0.20188
0.24955	0.04494	-0.01733	-0.15570	0.16051
0.30442	0.03499	-0.00465	-0.14054	0.11709
0.37056	0.02634	-0.00160	-0.14319	0.04513
0.43057	0.01737	-0.00364	-0.15402	-0.04915
0.49000	0.00739	-0.01950	-0.13798	-0.20660

UPPER CP INPUT

-0.214	-1.000	-6.940	-1.015	-1.020	-1.049	-1.081	-0.583
-0.517	-6.300	-0.271	-0.219	-0.154	-0.100	-0.041	

	LWLS	CP	INPUT					
-0.670	-0.000	-0.630	-0.620	-0.611	-0.561	-0.305	0.040	
0.211	0.410	0.404	0.401	0.531	0.547	0.570		
ITERATION	10	C1H	-0.49987 DPM	= 0.0093113H	AT 10	2 NSSP	15 DELTAY OR DELSTAR	= 0.0
ITERATION	20	C1H	-0.26461 DPM	= 0.00303018	AT 6	2 NSSP	14 DELTAY OR DELSTAR	= 0.0
ITERATION	30	C1H	-0.24985 DPM	= 0.0010190U	AT 24	3 NSSP	15 DELTAY OR DELSTAR	= 0.0
ITERATION	40	C1H	-0.31067 DPM	= 0.00027472U	AT 10	2 NSSP	15 DELTAY OR DELSTAR	= 0.0
ITERATION	50	C1H	-0.31011 DPM	= 0.00010498U	AT 11	3 NSSP	14 DELTAY OR DELSTAR	= 0.0222
ITERATION	60	C1H	-0.31009 DPM	= 0.00009540U	AT 14	3 NSSP	14 DELTAY OR DELSTAR	= 0.0022
ITERATION	70	C1H	-0.31010 DPM	= 0.00008621U	AT 10	2 NSSP	14 DELTAY OR DELSTAR	= 0.0073
ITERATION	80	C1H	-0.31012 DPM	= 0.00008204D	AT 15	3 NSSP	14 DELTAY OR DELSTAR	= 0.0037
ITERATION	90	C1H	-0.31011 DPM	= 0.00007914U	AT 10	3 NSSP	14 DELTAY OR DELSTAR	= 0.0025
ITERATION	100	C1H	-0.31010 DPM	= 0.00007614U	AT 11	3 NSSP	14 DELTAY OR DELSTAR	= 0.0018
ITERATION	110	C1H	-0.31010 DPM	= 0.00007301A	AT 11	3 NSSP	14 DELTAY OR DELSTAR	= 0.0013
ITERATION	120	C1H	-0.31010 DPM	= 0.00006979A	AT 10	5 NSSP	14 DELTAY OR DELSTAR	= 0.0010
ITERATION	130	C1H	-0.31010 DPM	= 0.0000640121	AT 10	4 NSSP	14 DELTAY OR DELSTAR	= 0.0007
ITERATION	140	C1H	-0.31010 DPM	= 0.00005949D	AT 11	5 NSSP	14 DELTAY OR DELSTAR	= 0.0005
ITERATION	150	C1H	-0.31010 DPM	= 0.000052194	AT 10	3 NSSP	14 DELTAY OR DELSTAR	= 0.0003
ITERATION	160	C1H	-0.31010 DPM	= 0.0000490113	AT 12	6 NSSP	14 DELTAY OR DELSTAR	= 0.0003
ITERATION	170	C1H	-0.31010 DPM	= 0.000041162	AT 10	3 NSSP	14 DELTAY OR DELSTAR	= 0.0002
ITERATION	180	C1H	-0.31010 DPM	= 0.000036657	AT 11	4 NSSP	14 DELTAY OR DELSTAR	= 0.0002
ITERATION	190	C1H	-0.31010 DPM	= 0.000031770	AT 4	5 NSSP	14 DELTAY OR DELSTAR	= 0.0001
ITERATION	200	C1H	-0.31010 DPM	= 0.000026542	AT 22	6 NSSP	14 DELTAY OR DELSTAR	= 0.0001
ITERATION	210	C1H	-0.31010 DPM	= 0.000020361	AT 12	3 NSSP	14 DELTAY OR DELSTAR	= 0.0000
ITERATION	220	C1H	-0.31010 DPM	= 0.000017767	AT 9	4 NSSP	14 DELTAY OR DELSTAR	= 0.0001
ITERATION	230	C1H	-0.31010 DPM	= 0.000014023	AT 16	2 NSSP	14 DELTAY OR DELSTAR	= 0.0000
ITERATION	240	C1H	-0.31010 DPM	= 0.000010378	AT 24	6 NSSP	14 DELTAY OR DELSTAR	= 0.0000
ITERATION	250	C1H	-0.31010 DPM	= 0.000007791	AT 21	10 NSSP	14 DELTAY OR DELSTAR	= 0.0000
ITERATION	260	C1H	-0.31010 DPM	= 0.000005239	AT 8	8 NSSP	14 DELTAY OR DELSTAR	= 0.0000
ITERATION	270	C1H	-0.31010 DPM	= 0.000003607	AT 24	6 NSSP	14 DELTAY OR DELSTAR	= 0.0000
ITERATION	280	C1H	-0.31010 DPM	= 0.000002157	AT 23	1 NSSP	14 DELTAY OR DELSTAR	= 0.0000
ITERATION	290	C1H	-0.31010 DPM	= 0.000001015	AT 23	0 NSSP	14 DELTAY OR DELSTAR	= 0.0000
ITERATION	300	C1H	-0.31010 DPM	= 0.000000494	AT 24	7 NSSP	14 DFTAY OR DELSTAR	= 0.0000
ITERATION	320	C1H	-0.31010 DPM	= 0.000000245	AT 24	6 NSSP	14 DFTAY OR DELSTAR	= 0.0000
ITERATION	340	C1H	-0.31010 DPM	= 0.0000001072	AT 22	7 NSSP	14 DFTAY OR DELSTAR	= 0.0000
ITERATION	360	C1H	-0.31010 DPM	= 0.0000000464	AT 24	7 NSSP	14 DFTAY OR DELSTAR	= 0.0000
ITERATION	380	C1H	-0.31010 DPM	= 0.0000000207	AT 24	6 NSSP	14 DFTAY OR DELSTAR	= 0.0000
ITERATION	400	C1H	-0.31010 DPM	= 0.0000000105	AT 23	0 NSSP	14 DFTAY OR DELSTAR	= 0.0000
ITERATION	420	C1H	-0.31010 DPM	= 0.0000000041	AT 23	0 NSSP	14 DFTAY OR DELSTAR	= 0.0000
ITERATION	440	C1H	-0.31010 DPM	= 0.0000000017	AT 24	7 NSSP	14 DFTAY OR DELSTAR	= 0.0000
ITERATION	460	C1H	-0.31010 DPM	= 0.0000000007	AT 24	6 NSSP	14 DFTAY OR DELSTAR	= 0.0000
ITERATION	480	C1H	-0.31010 DPM	= 0.0000000003	AT 23	7 NSSP	14 DFTAY OR DELSTAR	= 0.0000
ITERATION	500	C1H	-0.31010 DPM	= 0.0000000001	AT 23	7 NSSP	14 DFTAY OR DELSTAR	= 0.0000

## CH FY BACKWARD DIFFERENCES

X	CPU	CPL
-0.490	-0.200	0.502
-0.491	-1.280	-0.324
-0.492	-0.915	-0.087
-0.493	-1.007	-0.600
-0.494	-1.291	-0.631
-0.495	-1.016	-0.671
-0.496	-1.031	-0.641
-0.497	-1.030	-0.500
0.0	-1.051	-0.306
0.100	-0.284	0.37
0.125	-0.357	0.213
0.150	-0.311	0.214
0.175	-0.370	0.407
0.200	-0.311	0.401
0.225	-0.319	0.522
0.250	-0.130	0.501
0.275	-0.013	0.006
0.300	0.102	0.006
1.410	0.036	0.036
3.470	0.195	0.009

## CP FY CENTRAL DIFFERENCES

X	CPU	CPL
-0.490	-0.208	0.502
-0.491	-1.280	-0.324
-0.492	-0.918	-0.087
-0.493	-1.006	-0.600
-0.494	-1.287	-0.631
-0.495	-1.014	-0.671
-0.496	-1.030	-0.641
-0.497	-1.030	-0.500
0.0	-1.050	-0.306
0.100	-0.284	0.37
0.125	-0.357	0.213
0.150	-0.311	0.214
0.175	-0.370	0.407
0.200	-0.311	0.401
0.225	-0.319	0.522
0.250	-0.130	0.501
0.275	-0.013	0.006
0.300	0.102	0.006
1.410	0.036	0.036
3.472	0.004	0.004

## CP FY CENTRAL DIFFERENCES

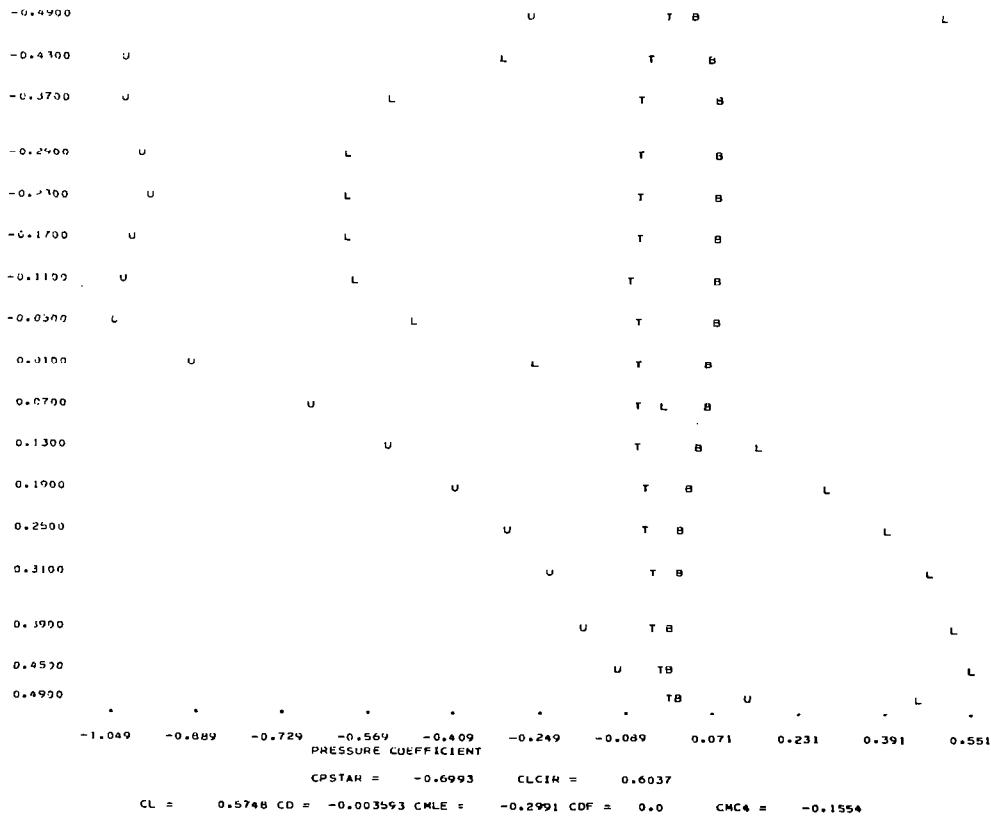
X	CPU	CPL
-0.490	-0.268	0.502
-0.491	-1.020	-0.324
-0.492	-1.018	-0.513
-0.493	-1.006	-0.600
-0.494	-1.287	-0.631
-0.495	-1.014	-0.671
-0.496	-1.030	-0.641
-0.497	-1.030	-0.500
0.0	-1.050	-0.306
0.100	-0.284	0.37
0.125	-0.357	0.213
0.150	-0.311	0.214
0.175	-0.370	0.407
0.200	-0.311	0.401
0.225	-0.319	0.522
0.250	-0.130	0.501
0.275	-0.013	0.006
0.300	0.102	0.006
1.410	0.036	0.036
3.472	0.004	0.004

X	YU	YL	SLL	SLL
-0.49000	0.02450	-0.02552	0.93718	-1.31772
-0.49067	0.04850	-0.06132	0.22258	-0.33099
-0.49038	0.05905	-0.07502	0.12977	-0.16525
-0.49082	0.06553	-0.08247	0.08442	-0.08434
-0.49052	0.06668	-0.08600	0.05405	-0.03140
-0.49077	0.07211	-0.08654	0.02741	0.01387
-0.49124	0.07306	-0.08418	0.00312	0.06265
-0.09240	0.07245	-0.07843	-0.02203	0.12524
0.0	0.07013	-0.08883	-0.05353	0.18729
0.06240	0.06395	-0.05144	-0.01944	0.21943
0.04570	0.05953	-0.04230	-0.11925	0.22409
0.18677	0.04233	-0.02912	-0.12857	0.16241
0.24852	0.04403	-0.01764	-0.13553	0.16710
0.30982	0.03566	-0.00891	-0.14042	0.11546
0.37058	0.02688	-0.00394	-0.14526	0.04496
0.43067	0.01790	-0.00414	-0.15427	-0.05072
0.49000	0.00815	-0.01211	-0.18591	-0.20427

71 71 71 71 71 71 71 71 71 71 71 71 71  
 70 69 69 69 c1 c1 c1 69 70 70 70 71  
 69 67 c5 63 c1 56 44 67 69 72 74  
 70 69 c5 55 c6 74 76 80 84 83  
 71 72 73 76 c3 010c 54 88 84 80  
 71 74 78 85 0 0112103 95 88 d1  
 71 76 82 89 c 01112166 99 91 d2  
 71 77 84 91 c 0 0107102 94 82  
 71 78 85 91 c 0 0109103 95 83  
 71 77 d3 90 c 0 0110104 96 83  
 71 76 d0 89 c 0 0111104 95 83  
 71 74 76 79 0 0 014 99 92 82  
 70 72 74 72 72 0 0 56 93 89 82  
 70 70 69 67 66 0 93 91 89 86 81  
 70 66 66 63 61 0 87 66 86 84 80  
 69 67 64 61 c6 0 62 43 c9 d2 80  
 69 66 62 54 55 0 60 60 80 80 79  
 69 67 61 58 54 0 74 73 73 74 76  
 69 65 62 50 54 0 75 75 76 77 77  
 67 61 57 54 53 0 68 71 74 77 79  
 67 67 67 67 c7 68 69 70 71 72 74  
 70 70 70 70 70 70 71 71 71 72  
 71 71 71 71 71 71 71 71 71 72

\*AVF CD = -0.003593

#### CHART 100



X-Y GRID SYSTEM

```

2 -0.8027E C1 = 3 -0.3872E 01 = 4 -0.2276E 01 = 5 -0.1410E 01 = 6 -0.4115E 00 = 7 -0.6471E 00
6 -0.5317E 00 = 9 -0.4900E 00 = 10 -0.4040E 00 = 11 -0.4307E 00 = 12 -0.4007E 00 = 13 -0.3706E 00
14 -0.3403E 00 = 15 -0.3056E 00 = 16 -0.2792E 00 = 17 -0.2485E 00 = 18 -0.2177E 00 = 19 -0.1868E 00
20 -0.1558E 00 = 21 -0.1247E 00 = 22 -0.9357E-01 = 23 -0.6240E-01 = 24 -0.3121E-01 = 25 0.0
26 0.3121E-C1 = 27 0.6240E-01 = 28 0.9357E-01 = 29 0.1247E 00 = 30 0.1558E 00 = 31 0.1868E 00
32 0.2177E 00 = 33 0.2485E 00 = 34 0.2792E 00 = 35 0.3098E 00 = 36 0.3403E 00 = 37 0.3706E 00
38 0.4040E 00 = 39 0.4307E 00 = 40 0.4604E 00 = 41 0.4900E 00 = 42 0.5317E 00 = 43 0.6471E 00
44 0.9115E 00 = 45 0.1410E 01 = 46 0.2276E 01 = 47 0.3872E 01 = 48 0.8027E 01 =
2 -0.1609E 01 = 3 -0.9181E 00 = 4 -0.5939E 00 = 5 -0.4261E 00 = 6 -0.3206E 00 = 7 -0.2460E 00
8 -0.1900E 00 = 9 -0.1420E 00 = 10 -0.1019E 00 = 11 -0.6592E-01 = 12 -0.3239E-01 = 13 -0.9213E-07
14 0.4239E-C1 = 15 0.6592E-01 = 16 0.1019E 00 = 17 0.1420E 00 = 18 0.1888E 00 = 19 0.2460E 00
20 0.3200E 00 = 21 0.4261E 00 = 22 0.5939E 00 = 23 0.9181E 00 = 24 0.1609E 01 =

```

MACH NO= 1.0720 ANGLE OF ATTACK IS 0.0 DEGREES  
DIRECT SOLUTION TO = -0.38  
CASE NUMBER 100

INVERSE DESIGN CASE

```

DEFINP
M= 0.71449997 ,N= 1.6999998 ,X1=-0.3H000000 ,X2= 0.5H000000 ,AL= 0.7 ,EPS= 0.0 ,EPSC= 0.0
0.39999998 ,X4= 0.48999995 ,X5= 2.0000000 ,CONV= 0.99999943E-06 ,AI= 0.245199999 ,AJ= 3.8699999
RN= 2999999999 ,XIBDLY= 0.34799999 ,CIR= 0.30105909 ,LCODEMH= 0.0 ,KDEL= 0.25000000 ,KDELFN= 0.12500000
SP= 0.39999999E-02 ,XSEP= 0.459999998 ,HCIRD= 0.199999999 ,LPB= 0.349999998 ,LMUNF= 0.46999997 ,XLSEP= 0.50000000 ,XPC=
0.99999964E-C1
END
CLINP
IMAX= 49,JMAX= 25,IKSEL= 100,ILNV= 1,MITER= 400,NHALF= 2,ITACT= 1000,ITEUPC= 0,ISKP2=
0,ISKP3= 0,ISKP4= 0,ITERP= 0,IRFAU= 0,LP= 1000,LTU= 0,ITELWC=
0
END

```

AIRFIELD COORDINATES

X	YL	YL	UPPER SLOPE LOWER SLOPE
-0.494000	0.02458	-0.02952	0.93718 -1.31772
-0.46044	0.04034	-0.04997	0.38482 -0.52526
-0.43067	C.04856	-0.08132	0.22298 -C.33099
-0.40072	C.05427	-0.08950	0.16531 -C.22929

UPPER CP INPUT

-1.107	-1.101	-1.058	-1.067	-1.076	-1.078	-1.069	-1.061
-1.057	-1.056	-1.055	-1.017	-0.917	-0.808	-0.701	-0.608
-0.526	-C.438	-0.379	-0.326	-0.204	-0.244	-0.207	-0.162
-0.144	-0.114	-0.087	-0.055	-0.007			

LOWER CP INPUT

-0.673	-C.677	-0.655	-0.657	-0.649	-0.637	-0.650	-0.628
-0.621	C.690	-0.528	-0.499	-0.267	-0.139	0.002	0.199
0.167	0.241	0.292	0.357	0.396	0.436	0.470	0.498
0.516	C.528	0.531	0.510	0.534			

```

ITERATION 10 CIR = 0.30204 DPM = 0.000127091 AT 39 17 NSSP = 67 DELTAY OR DELSTAR = 0.0
ITERATION 20 CIR = 0.31161 DPM = 0.000184659 AT 42 13 NSSP = 68 DELTAY OR DELSTAR = 0.0
ITERATION 30 CIR = 0.30163 DPM = 0.000055687 AT 35 2 NSSP = 67 DELTAY OR DELSTAR = 0.0
ITERATION 40 CIR = 0.30165 DPM = 0.00004399 AT 35 2 NSSP = 67 DELTAY OR DELSTAR = 0.0
ITERATION 50 CIR = 0.30163 DPM = 0.00003616 AT 36 2 NSSP = 67 DELTAY OR DELSTAR = 0.0
ITERATION 60 CIR = 0.30113 DPM = 0.00007592 AT 46 2 NSSP = 68 DELTAY OR DELSTAR = 0.0036
ITERATION 70 CIR = 0.30094 DPM = 0.00002682 AT 47 2 NSSP = 67 DELTAY OR DELSTAR = 0.0018
ITERATION 80 CIR = 0.30094 DPM = 0.00001264 AT 47 2 NSSP = 67 DELTAY OR DELSTAR = 0.0004
ITERATION 90 CIR = 0.30094 DPM = 0.000002229 AT 41 2 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 100 CIR = 0.30094 DPM = 0.000005334 AT 48 11 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 110 CIR = 0.30094 DPM = 0.000002979 AT 49 13 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 120 CIR = 0.30094 DPM = 0.000002983 AT 40 13 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 130 CIR = 0.30094 DPM = 0.000002005 AT 47 7 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 140 CIR = 0.30094 DPM = 0.000001738 AT 21 2 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 150 CIR = 0.30094 DPM = 0.000002336 AT 2 7 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 160 CIR = 0.30094 DPM = 0.000002994 AT 48 7 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 170 CIR = 0.30094 DPM = 0.000002834 AT 48 13 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 180 CIR = 0.30094 DPM = 0.000002784 AT 48 12 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 190 CIR = 0.30094 DPM = 0.000002900 AT 48 13 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 200 CIR = C.30094 DPM = 0.000002027 AT 44 12 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 210 CIR = 0.30094 DPM = 0.000002140 AT 46 11 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 220 CIR = 0.30094 DPM = 0.000002601 AT 47 12 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 230 CIR = 0.30094 DPM = 0.000003007 AT 48 13 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 240 CIR = 0.30094 DPM = 0.000002354 AT 2 7 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 250 CIR = 0.30094 DPM = 0.000002956 AT 48 8 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 260 CIR = 0.30094 DPM = 0.000001323 AT 44 3 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 270 CIR = 0.30094 DPM = 0.000002342 AT 46 9 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 280 CIR = 0.30094 DPM = 0.000003570 AT 47 12 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 290 CIR = 0.30094 DPM = 0.000001292 AT 47 13 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 300 CIR = 0.30094 DPM = 0.000003431 AT 48 13 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 310 CIR = 0.30094 DPM = 0.000002964 AT 48 7 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 320 CIR = 0.30094 DPM = 0.000001991 AT 47 12 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 330 CIR = C.30094 DPM = 0.000003204 AT 48 13 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 340 CIR = 0.30094 DPM = 0.000003159 AT 46 12 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 350 CIR = 0.30094 DPM = 0.000001216 AT 46 9 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 360 CIR = 0.30094 DPM = 0.000005831 AT 48 13 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 370 CIR = 0.30094 DPM = 0.000001198 AT 47 9 NSSP = 67 DELTAY OR DELSTAR = 0.0001
ITERATION 380 CIR = 0.30094 DPM = 0.000001224 AT 43 13 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 390 CIR = 0.30094 DPM = 0.000002964 AT 48 13 NSSP = 67 DELTAY OR DELSTAR = 0.0000
ITERATION 400 CIR = 0.30094 DPM = 0.000001361 AT 46 13 NSSP = 67 DELTAY OR DELSTAR = 0.0001

```

CP BY BACKWARD DIFFERENCES		
X	CPU	CPL
-0.490	-0.561	0.504
-0.480	-0.479	-0.111
-0.431	-1.276	-0.631
-0.401	-1.111	-0.584
-0.371	-1.108	-0.572
-0.340	-1.102	-0.677
-0.310	-1.009	-0.655
-0.279	-1.008	-0.655
-0.249	-1.077	-0.650
-0.218	-1.070	-0.648
-0.187	-1.070	-0.651
-0.156	-1.062	-0.629
-0.125	-1.068	-0.622
-0.094	-1.057	-0.590
-0.062	-1.051	-0.526
-0.031	-1.018	-0.408
0.0	-0.918	-0.267
0.031	-0.809	-0.139
0.062	-0.702	0.001
0.094	-0.609	0.088
0.125	-0.525	0.167
0.156	-0.439	0.241
0.187	-0.379	0.292
0.218	-0.326	0.357
0.249	-0.284	0.396
0.279	-0.284	0.436
0.310	-0.207	0.470
0.340	-0.164	0.498
0.371	-0.144	0.517
0.401	-0.114	0.529
0.431	-0.077	0.532
0.460	-0.055	0.511
0.490	-0.003	0.527
0.512	0.175	0.046
0.647	0.117	0.117
0.912	0.064	0.064
1.410	0.025	0.023
2.279	0.008	0.008
3.472	0.003	0.003
6.027	0.001	0.001
CP BY CENTRAL DIFFERENCES		
X	CPU	CPL
-0.490	-0.561	0.504
-0.480	-0.479	-0.111
-0.431	-1.081	-0.400
-0.401	-1.097	-0.535
-0.371	-1.103	-0.619
-0.340	-1.099	-0.684
-0.310	-1.066	-0.655
-0.279	-1.076	-0.653
-0.249	-1.076	-0.687
-0.218	-1.074	-0.693
-0.187	-1.068	-0.640
-0.156	-1.056	-0.625
-0.125	-1.063	-0.611
-0.094	-1.058	-0.573
-0.062	-1.043	-0.501
-0.031	-0.943	-0.393
0.0	-0.905	-0.202
0.031	-0.805	-0.135
0.062	-0.708	-0.023
0.094	-0.615	0.076
0.125	-0.527	0.101
0.156	-0.450	0.236
0.187	-0.366	0.297
0.218	-0.333	0.314
0.249	-0.287	0.394
0.279	-0.246	0.434
0.310	-0.206	0.468
0.340	-0.172	0.493
0.371	-0.143	0.514
0.401	-0.115	0.527
0.431	-0.084	0.527
0.460	-0.041	0.520
0.490	0.104	0.347
0.532	0.174	0.179
0.647	0.195	0.105
0.912	0.153	0.053
1.410	0.023	0.033
2.279	0.004	0.009
3.472	0.003	0.003
6.027	0.001	0.001



## CHART 100

-0.4900		U		T	B					
-0.4500	U	U	L	T	B					
-0.4300				T	B					
-0.3900	L	L		T	B					
-0.3700	U	L		T	B					
-0.3300	U	L		T	B					
-0.2900	U	L		T	B					
-0.2700	U	L		T	B					
-0.2300	U	L		T	B					
-0.2100	U	L		T	B					
-0.1700	U	L		T	B					
-0.1500	U	L		T	B					
-0.1100	U	L		T	B					
-0.0902	U	L		T	B					
-0.0500	U	U	L	T	B					
-0.0300			L	T	B					
0.0100	U		L	T	B					
0.0500	U	U	L	T	B					
0.0700			T	L	B					
0.1100		U		T	B					
0.1300		U		T	B					
0.1700		U		T	B					
0.1900		U		T	B					
0.2300		U		T	B					
0.2500		U		T	B					
0.2900		U		T	B					
0.3100		U		T	B					
0.3500		U		T	B					
0.3900		U		T	B					
0.4100		U		T	B					
0.4500		U		T	B					
0.4700		U		T	B					
0.4900		U		T	B					
.	.	.	.	.	.					
-1.103	-0.940	-0.777	-0.614	-0.451	-0.288	-0.125	0.038	0.201	0.364	0.527
PRESSURE COEFFICIENT										
CPSTAR = -0.6993 CLCIR = 0.6019										
CL = 0.5875 CD = -0.016110 CMLE = -0.2989 CDF = 0.0 CMCA = -0.1520										

## BOUNDARY LAYER ANALYSIS FOR HEYNOLDS NO. OF 0.210E 08

X	Y	YNEW	P	DELS	THETA	SEP	H	PI	TAU	
-0.34028	0.06227	0.06226	1.17571	0.00001	0.00000	0.98663	0.00612	12	0.00186	
-0.30982	0.06516	0.06503	1.16937	0.00014	0.00007	0.00001	1.94562	0.01163	13	0.00155
-0.27923	0.06750	0.06727	1.16482	0.00023	0.00012	0.00000	1.90204	-0.00089	12	0.00143
-0.24952	0.06539	0.06509	1.16504	0.00031	0.00016	0.00000	1.88197	0.00562	11	0.00136
-0.21769	0.07735	0.07047	1.16403	0.00038	0.00020	0.00001	1.87019	0.01942	11	0.00130
-0.18677	0.07191	0.07146	1.16128	0.00046	0.00025	0.00001	1.85981	0.01085	11	0.00126
-0.15577	0.07291	0.07208	1.16002	0.00053	0.00029	0.00001	1.85084	0.01224	10	0.00123
-0.12477	0.07293	0.07233	1.15879	0.00060	0.00032	0.00001	1.84566	0.02632	10	0.00120
-0.09357	0.07287	0.07220	1.15651	0.00067	0.00036	0.00004	1.84177	0.02922	10	0.00117
-0.06240	0.07243	0.07168	1.14975	0.00075	0.00040	0.00014	1.84754	0.37927	9	0.00110
-0.03121	0.07158	0.06973	1.14933	0.00083	0.00044	0.00013	1.85097	0.81090	10	0.00101
0.0	0.07634	0.06935	1.108770	0.00098	0.00052	0.00002	1.86014	1.16056	12	0.00095
0.03121	0.06863	0.06741	1.04429	0.00111	0.00061	0.00004	1.84241	1.36936	12	0.00092
0.06240	0.06611	0.06481	1.00333	0.00126	0.00069	0.00073	1.81897	1.57480	12	0.00086
0.09357	0.06304	0.06160	0.96512	0.00142	0.00079	0.00082	1.80016	1.82663	13	0.00085
0.12477	0.05873	0.05802	0.92514	0.00159	0.00090	0.00087	1.77761	1.84233	13	0.00084
0.15577	0.05545	0.05408	0.90825	0.00175	0.00100	0.00086	1.74630	1.72344	12	0.00084
0.18677	0.05146	0.04995	0.87277	0.00190	0.00111	0.00062	1.71513	1.60013	12	0.00084
0.21769	0.04773	0.04664	0.85120	0.00201	0.00120	0.00077	1.68841	1.49107	12	0.00085
0.24952	0.04351	0.04132	0.83348	0.00217	0.00130	0.00075	1.66064	1.47861	12	0.00084
0.27923	0.03922	0.03688	0.81730	0.00232	0.00140	0.00078	1.63934	1.61199	12	0.00083
0.30982	0.03488	0.03239	0.80160	0.00247	0.00150	0.00079	1.64965	1.52716	11	0.00083
0.34024	0.03090	0.02788	0.79786	0.00259	0.00159	0.00073	1.62724	1.31734	11	0.00084
0.37058	0.02695	0.02330	0.77658	0.00272	0.00168	0.00071	1.61511	1.44769	11	0.00083
0.40072	0.02194	0.01863	0.76533	0.00280	0.00178	0.00080	1.62124	1.75212	11	0.00080
0.43067	0.01712	0.01379	0.75330	0.00291	0.00186	0.00109	1.66644	3.26076	12	0.00069
0.46094	0.01238	0.00959	0.73642	0.00292	0.00202	0.00373	2.47992*****	*****	25	0.00000
0.49000	0.00692	-0.00208	0.78334	0.00293	0.00290	0.00861	3.24841*****	*****	26	0.00000
X	YLLD	YNEW	DELSTAR							
-0.46344	0.04314	0.04034	0.0							
-0.43007	0.04656	0.04495	0.0							
-0.40072	0.05127	0.05427	0.00000							
-0.37058	0.05571	0.05571	0.00001							
-0.34024	0.06227	0.06122	0.00005							
-0.30942	0.06716	0.06503	0.00013							
-0.27923	0.07170	0.06727	0.00022							
-0.24952	0.07639	0.06909	0.00030							
-0.21769	0.07745	0.07047	0.00036							
-0.18677	0.07711	0.07146	0.00046							
-0.15577	0.07201	0.07201	0.00053							
-0.12477	0.07293	0.07213	0.00060							
-0.09357	0.07287	0.07220	0.00067							
-0.06240	0.07243	0.07167	0.00076							
-0.03121	0.07158	0.07072	0.00086							
0.0	0.07633	0.06943	0.00096							
0.03121	0.06633	0.06741	0.00112							
0.06240	0.06111	0.06463	0.00127							
0.09357	0.06105	0.06105	0.00143							
0.12477	0.05563	0.05803	0.00156							
0.15577	0.05573	0.05469	0.00174							
0.18677	0.05156	0.04995	0.00194							
0.21769	0.04773	0.04643	0.00203							
0.24952	0.04351	0.04131	0.00217							
0.27923	0.03922	0.03688	0.00232							
0.30982	0.03488	0.03240	0.00246							
0.34024	0.03090	0.02788	0.00259							
0.37058	0.02695	0.02330	0.00274							
0.40072	0.02194	0.01849	0.00302							
0.43067	0.01711	0.01391	0.01191							
0.46094	0.01238	0.00950	0.00911							
0.49000	0.00692	-0.00208	0.00953	0.00290	0.00861	3.24841*****	*****	26	0.00000	

## BLINDARY LAYER ANALYSIS FOR REYNOLDS NO. LF 0.210E 08

X	Y	YNEW	M	DELM	THETA	SUP	M	P1	TAU
-0.34026	-0.7569	-0.07400	0.97846	0.00001	0.00001	-0.00000	0.90427	-0.00294	12 0.00192
-0.33742	-0.68276	-0.06203	0.98148	0.00014	0.00007	-0.00000	1.70243	0.00260	13 0.00160
-0.27923	-0.68449	-0.06460	0.98060	0.00021	0.00012	0.00001	1.74446	0.01190	12 0.00148
-0.24452	-0.68021	-0.06552	0.97804	0.00029	0.00017	0.00001	1.72540	0.00574	11 0.00140
-0.21769	-0.68032	-0.06046	0.97719	0.00036	0.00021	0.00001	1.71315	0.01401	11 0.00134
-0.18677	-0.68074	-0.06631	0.97549	0.00043	0.00025	0.00002	1.70517	0.04544	10 0.00130
-0.15577	-0.68133	-0.06543	0.97049	0.00050	0.00030	0.00003	1.69911	0.00937	10 0.00125
-0.12470	-0.68447	-0.06839	0.96359	0.00054	0.00034	0.00011	1.65683	0.23219	8 0.00119
-0.09457	-0.68156	-0.06829	0.96788	0.00067	0.00039	0.00026	1.70275	0.35666	10 0.00111
-0.06240	-0.68667	-0.07778	0.91864	0.00079	0.00046	0.00051	1.71805	1.14462	12 0.00101
-0.03121	-0.68717	-0.07117	0.97553	0.00096	0.00055	0.00085	1.74244	2.06364	14 0.00093
0.2	-0.68674	-0.06750	0.98234	0.00120	0.00068	0.00119	1.70345	2.94553	14 0.00090
0.03121	-0.68126	-0.06100	0.77351	0.00149	0.00094	0.00144	1.77323	3.76564	15 0.00073
0.36249	-0.65588	-0.05401	0.72909	0.00187	0.00102	0.00166	1.76474	4.80806	15 0.00066
0.09357	-0.64943	-0.04966	0.68969	0.00221	0.00123	0.00197	1.79956	5.81726	15 0.00061
0.11470	-0.64193	-0.04924	0.65553	0.00266	0.00146	0.00202	1.82415	7.49914	16 0.00054
0.15577	-0.63110	-0.03194	0.64424	0.00315	0.00172	0.00222	1.83651	7.40788	16 0.00053
0.16677	-0.61772	-0.02902	0.59963	0.00366	0.00198	0.00223	1.83148	8.56462	15 0.00050
0.31769	-0.62026	-0.01845	0.57777	0.00415	0.00225	0.00240	1.84576	9.98870	10 0.00046
0.24652	-0.61730	-0.01240	0.55824	0.00474	0.00254	0.00253	1.86721	11.32496	17 0.00043
0.27923	-0.61926	-0.00704	0.54025	0.00531	0.00284	0.00256	1.87079	11.39692	16 0.00042
0.30627	-0.60911	-0.00264	0.52680	0.00573	0.00314	0.00247	1.86202	8.46838	12 0.00048
0.34723	-0.600136	-0.00054	0.51372	0.00588	0.00340	0.00212	1.72612	5.45117	10 0.00058
0.37511	-0.600349	-0.00242	0.50458	0.00580	0.00362	0.00159	1.60312	2.81727	11 0.00070
0.40972	-0.60291	-0.00281	0.49746	0.00546	0.00377	0.00067	1.45314	0.09675	14 0.00094
0.44057	-0.60534	-0.00169	0.44949	0.00447	0.00349	-0.00037	1.30921	-0.88923	13 0.00113
0.46044	-0.60669	-0.00194	0.50252	0.00457	0.00376	-0.00611	1.21603	-1.30300	12 0.00144
0.49400	-0.61147	-0.00723	0.56114	0.00316	0.00281	-0.01676	1.12255	-1.50000	12 0.00143
X	YUL11	YNLW	YELSTAR						
-0.46044	-0.64817	-0.04057	0.0						
-0.43007	-0.66112	-0.06132	0.0						
-0.40972	-0.66930	-0.06750	0.00000						
-0.37511	-0.67543	-0.07592	0.00001						
-0.34026	-0.67505	-0.07946	0.00005						
-0.30482	-0.67276	-0.08264	0.00012						
-0.27923	-0.67449	-0.08466	0.00021						
-0.24652	-0.67021	-0.08959	0.00029						
-0.21769	-0.67462	-0.09046	0.00036						
-0.19677	-0.67674	-0.09630	0.00043						
-0.15577	-0.68593	-0.08543	0.00151						
-0.12470	-0.68417	-0.08379	0.00059						
-0.09357	-0.68146	-0.08127	0.00069						
-0.06240	-0.67637	-0.07775	0.00082						
-0.03121	-0.67415	-0.07314	0.00099						
0.0	-0.67674	-0.06750	0.00123						
0.33121	-0.68078	-0.06104	0.00151						
0.30627	-0.68530	-0.05370	0.00165						
0.28057	-0.68482	-0.04663	0.00224						
0.24652	-0.68170	-0.03972	0.00268						
0.15577	-0.68316	-0.03194	0.00315						
0.13077	-0.68272	-0.02950	0.00366						
0.11764	-0.68101	-0.01843	0.00418						
0.09557	-0.67120	-0.01481	0.00473						
0.07223	-0.67240	-0.00711	0.00523						
0.04997	-0.66641	-0.00277	0.00566						
0.03470	-0.66036	-0.00042	0.00576						
0.01705	-0.66339	-0.00230	0.00568						
0.00672	-0.66743	-0.00176	0.00539						
0.40407	-0.67129	-0.01165	0.00493						
0.42244	-0.67554	-0.00142	0.00420						
0.49400	-0.61647	-0.00654	0.00346						

CDF = 0.70748  
NO UPFLR SURFACE SEPARATION BEFORE 0.46000

Sample Case No. 2 - Airfoil Analysis with Viscous Interaction

SAMPLE CASE NO. P--KCRN AIRFCIL 75-06-12 TWO GRID HALVINGS ONLY

&INP M=0.752, RN=20.95E+06,&END

&IINP ITACT=1,&END

61

0.0	0.00074	0.00003	0.00399	0.00056	0.00717	0.00171	0.01015
0.00365	0.01296	0.00640	0.01573	0.00993	0.01852	0.01415	0.02133
0.01906	0.02415	0.02461	0.02696	0.03080	0.02973	0.03761	0.03245
0.04503	0.03510	0.05304	0.03766	0.06163	0.04009	0.07080	0.04237
0.08057	0.04444	0.09101	0.04629	0.10215	0.04800	0.11398	0.04963
0.12642	0.05118	0.13946	0.05266	0.15306	0.05408	0.16720	0.05543
0.18183	0.05670	0.19694	0.05791	0.21251	0.05903	0.22850	0.06007
0.24488	0.06104	0.27873	0.06271	0.31385	0.06403	0.35000	0.06499
0.38695	0.06556	0.42446	0.06575	0.46229	0.06553	0.50019	0.06489
0.53793	0.06379	0.57526	0.06219	0.61198	0.06003	0.64797	0.05727
0.68302	0.05390	0.71710	0.04996	0.75007	0.04552	0.78179	0.04069
0.81214	0.03558	0.84096	0.03032	0.86810	0.02508	0.89338	0.02003
0.91660	0.01533	0.93749	0.01114	0.95580	0.00759	0.96390	0.00607
0.97125	0.00472	0.97783	0.00355	0.98360	0.00256	0.98854	0.00175
0.99262	0.00109	0.99582	0.00060	0.99813	0.00026	0.99953	0.00006
1.00000	0.00000						
0.0	1.0	0.0	0.0				

51

0.00000	0.00074	0.00049	-0.00240	0.00165	-0.00538	0.00351	-0.00827
0.00605	-0.01105	0.00932	-0.01370	0.01336	-0.01626	0.01814	-0.01877
0.02366	-0.02122	0.02087	-0.02364	0.03676	-0.02602	0.04431	-0.02834
0.05250	-0.03061	0.07074	-0.03491	0.09134	-0.03886	0.11420	-0.04237
0.13920	-0.04538	0.16622	-0.04789	0.19512	-0.04978	0.22574	-0.05112
0.25793	-0.05186	0.27455	-0.05200	0.30873	-0.05181	0.34401	-0.05097
0.38020	-0.04946	0.41714	-0.04726	0.45467	-0.04436	0.49263	-0.04077
0.53091	-0.03653	0.56937	-0.03168	0.60792	-0.02634	0.64646	-0.02065
0.68491	-0.01485	0.72309	-0.00930	0.74198	-0.00675	0.76068	-0.00442
0.79713	-0.00059	0.83177	-0.00206	0.86398	-0.00362	0.89331	0.00428
0.90679	0.00435	0.91943	0.00427	0.93122	0.00407	0.95212	0.00342
0.96934	0.00255	0.98278	0.00163	0.99237	0.00080	0.99572	0.00047
0.99810	0.00022	0.99953	0.00006	1.00000	0.00000		
0.0	-1.0	0.0	0.0				

SAMPLE CASE NO. 2--KORN AIRFOIL 75-06-12 TWO GRID HALVINGS ONLY

X-Y GRID SYSTEM

2 -0.1410E 01 3 -0.4900E 00 4 -0.3706E 00 5 -0.2485E 00 6 -0.1247E 00 7 0.0  
8 0.1247E 00 9 0.2485E 00 10 0.3706E 00 11 0.4900E 00 12 0.1410E 01  
2 -0.4261E 00 3 -0.1420E 00 4 -0.2303E-07 5 0.1420E 00 6 0.4261E 00

MACH NO= 15 0.752 ANGLE OF ATTACK IS 0.0 DEGREES  
DIRECT SOLUTION TO 0.50  
CASE NUMBER 100

INVIScid ANALYSIS CASE  
WITH Viscous INTERACTION

DEFINP  
M= 0.75199997 ,W= 1.0999998 ,X1= 0.50000000 ,X2= 1.0000.000 ,ALP= 0.0 ,EPS= 0.0 ,EPSS= 0.8699999  
0.39999998 ,X4= 0.49999995 ,S4= 2.0000000 ,CONV= 0.99999997E-05,A1= 0.24599999 ,A2= 0.14999998 ,A3= 3.8699999  
RN= 20950000. ,XIBOLY= 0.444000000 ,CIR= 0.0 ,CDCORR= 0.0 ,RDEL= 0.25000000 ,RDELFN= 0.12500000  
SP= 0.39999996E-02,XSEP= 0.440000000 ,RCPB= 0.19999999 ,CPB= 0.39999998 ,XMDN= 0.46999997 ,XLSEP= 0.50000000 ,XPC= 0.99999964E-01  
END  
CINP  
IMAX= 13,JMAX= 7,IKASE= 100,INV= 0,ITERP= 0,ITERAD= 0,LP= 1000,ITEUPC= 0,ITELWC= 0  
EEND

AIRFOIL COORDINATES

X	YU	YL	UPPER SLOPE	LOWER SLOPE
-0.49000	0.01857	-0.01417	0.71872	-0.67988
-0.37058	0.05153	-0.04420	0.11615	-0.11678
-0.24852	0.06140	-0.05176	0.05326	-0.01602
-0.12470	0.06552	-0.04970	0.01340	0.04823
0.0	0.06499	-0.04000	-0.02273	0.11785
0.12470	0.05913	-0.02390	-0.07384	0.14744
0.24852	0.04574	-0.00591	-0.14295	0.12644
0.37058	0.02459	0.00383	-0.19774	0.02909
0.49000	0.00151	0.00102	-0.16299	-0.09073

ITERATION 10 CIR = 0.09421 DPM = 0.00389451 AT 4 3 NSSP = 0 DELTAY OR DELSTAR = 0.0  
ITERATION 20 CIR = 0.13322 DPM = 0.003370694 AT 12 3 NSSP = 1 DELTAY OR DELSTAR = 0.0  
ITERATION 30 CIR = 0.14425 DPM = 0.00196538 AT 10 3 NSSP = 3 DELTAY OR DELSTAR = 0.0  
ITERATION 40 CIR = 0.16120 DPM = 0.00168639 AT 10 3 NSSP = 4 DELTAY OR DELSTAR = 0.0  
ITERATION 50 CIR = 0.17540 DPM = 0.00143486 AT 10 3 NSSP = 4 DELTAY OR DELSTAR = 0.0

CP BY CENTRAL DIFFERENCES

X CPU CPU

X	CPU	CPU
-0.490	0.001	0.280
-0.371	-0.947	-0.791
-0.249	-0.722	-0.492
-0.125	-0.679	-0.529
0.0	-0.705	-0.324
0.125	-0.659	-0.047
0.249	-0.444	0.238
0.371	-0.083	0.376
0.490	0.570	0.651
1.410	0.031	0.030

X	YU	YL	SLU	SLL
-0.49000	0.01857	-0.01417	0.71872	-0.67988
-0.37058	0.05153	-0.04420	0.11615	-0.11678
-0.24852	0.06140	-0.05176	0.05325	-0.01692
-0.12470	0.06552	-0.04070	0.01340	0.04823
0.0	0.06499	-0.04000	-0.02273	0.10586
0.12470	0.05913	-0.02390	-0.07384	0.14744
0.24852	0.04574	-0.00591	-0.14295	0.12644
0.37058	0.02459	0.00383	-0.19774	0.02909
0.49000	0.00151	0.00102	-0.16299	-0.09073

74 75 72 74 75  
68 62 0 71 75  
80 95 0 0101 E5  
83 97 0 0101 S0  
83 92 0 0101 S3  
80 85 0 0102 S2  
76 77 0 99 E5  
73 69 0 90 E4  
70 65 0 79 78  
55 49 0 56 6E  
74 73 73 74 74

WAVE CD = -0.002310

## CHART 100

-0.4900			TUB	L							
-0.3700	U	L	T	B							
-0.2300	UL		T	B							
-0.1100	U	L	T	B							
0.0100	U	L	T	B							
0.1300	U		TL	B							
0.2500	U		T	B	L						
0.3900			U	T	B	L					
0.4900				B		U	L				
-0.947	-0.787	-0.627	-0.468	-0.308	-0.148	0.012	0.172	0.332	0.492	0.651	
PRESSURE COEFFICIENT											
CPSTAR = -0.5842 CLCIR = 0.3508											
CL = 0.3275 CD = -0.002310 CMLE = -0.1982 CDF = 0.0 CMC4 = -0.1163											

SAMPLE CASE NO. 2-KOHN AIRFOIL 78-0612 TWO GRID HALVINGS ONLY

X-Y GRID SYSTEM

X	Y	X	Y	X	Y	X	Y
-8.4307E-01	3.0.1410E-01	4.0.6471E-00	5.0.4990E-00	6.0.4307E-00	7.0.3706E-00		
-8.4304E-00	0.0.1408E-00	0.0.1408E-00	0.0.1408E-00	0.0.1408E-00	0.0.1408E-00	0.0.1408E-00	
0.0.2402E-01	1.0.1207E-00	1.0.1207E-00	1.0.1207E-00	1.0.1207E-00	1.0.1207E-00	1.0.1207E-00	
0.0.3075E-00	2.0.04900E-00	2.0.04900E-00	2.0.04900E-00	2.0.04900E-00	2.0.04900E-00	2.0.04900E-00	
0.0.4594E-00	3.0.04420E-00	3.0.04420E-00	3.0.04420E-00	3.0.04420E-00	3.0.04420E-00	3.0.04420E-00	
0.0.6594E-01	4.0.02400E-00	4.0.02400E-00	4.0.02400E-00	4.0.02400E-00	4.0.02400E-00	4.0.02400E-00	
0.0.9459E-01	5.0.01420E-00	5.0.01420E-00	5.0.01420E-00	5.0.01420E-00	5.0.01420E-00	5.0.01420E-00	
0.0.1408E-01	6.0.01420E-00	6.0.01420E-00	6.0.01420E-00	6.0.01420E-00	6.0.01420E-00	6.0.01420E-00	

MACH NO. IS 0.702 ANGLE OF ATTACK IS 0.0 DEGREES

DIRECT SOLUTION TO 0.50

CASE NUMBER 100

INVIScid ANALYSIS CASE  
WITH VISCCS INTERACTION

```
CFINP
M= 0.75199997 .XN 1.65999998 .X1 0.30000000 .X2 0.10000000 .ALPA 0.0 .EPS= 0.0 .EPSS=
0.39999998 .XAN 0.48999995 .SN 2.00000000 .CDHVS= 0.99999997E-05 .A1= 0.24599999 .A2= 0.14999998 .A3= 3.86999999
RHO 20000000 .XISOLY= 0.44000000 .CIRL= 0.17540183 .CDCDRH= 0.0 .RDEL= 0.25000000 .RDELFC= 0.12500000
SPL 0.39999998E-02 .XSEP= 0.44000000 .RCFB= 0.19999999 .CPB= 0.39999998 .XMDN= 0.46999997 .XLSEP= 0.50000000 .XPC=
0.99999984E-01
DEND
FINP
IMAX= 25 .JMAX= 13 .IMASEN= 100 .INW= 0 .ITER= 0 .NITER= 0 .NHALF= 2 .ITACT= 0 .ISKP2=
0 .ISKP3= 0 .ISKP4= 0 .LPR= 1000 .TEUPC= 0 .TELVC=
```

DEND

AIRFOIL COORDINATES

X	Y	TL	UPPER SLOPE	LOWER SLOPE
-0.49000	0.011657	-0.014117	0.71872	-0.67988
-0.43047	0.012023	-0.03461	0.23468	-0.21554
-0.37058	0.015153	-0.04420	0.11615	-0.11678
-0.30962	0.016736	-0.04951	0.07986	-0.05735
-0.24852	0.016140	-0.05176	0.05325	-0.01692
-0.18677	0.016461	-0.05176	0.03230	0.01706
-0.12470	0.016542	-0.04970	0.01346	0.04823
-0.06240	0.016572	-0.04870	-0.00416	0.07808
0.0	0.016489	-0.04808	-0.02273	0.10506
0.06240	0.016280	-0.03260	-0.04500	0.13057
0.12470	0.015913	-0.02390	-0.07386	0.14744
0.18677	0.015350	-0.01457	-0.10807	0.14944
0.24852	0.014574	-0.00591	-0.14293	0.12064
0.30962	0.013599	-0.00956	-0.17483	0.06037
0.37058	0.012559	-0.03383	-0.19774	0.02909
0.43047	0.012556	-0.04846	-0.19995	-0.02139
0.49000	0.010151	-0.01012	-0.16289	0.09073

ITERATION 10 CIR = 0.185802 DPM = 0.00280645d AT 12 3 NSSP = 19 DELTAY OR DELSTAR = 0.0
ITERATION 20 CIR = 0.19592 DPM = 0.00099790 AT 24 6 NSSP = 17 DELTAY OR DELSTAR = 0.0
ITERATION 30 CIR = 0.20530 DPM = 0.00087315 AT 24 6 NSSP = 17 DELTAY OR DELSTAR = 0.0
ITERATION 40 CIR = 0.21345 DPM = 0.00076777 AT 24 6 NSSP = 17 DELTAY OR DELSTAR = 0.0
ITERATION 50 CIR = 0.22063 DPM = 0.00070266 AT 22 4 NSSP = 17 DELTAY OR DELSTAR = 0.0
ITERATION 60 CIR = 0.21917 DPM = 0.00061113 AT 15 6 NSSP = 16 DELTAY OR DELSTAR = 0.0057
ITERATION 70 CIR = 0.21802 DPM = 0.00046831 AT 15 6 NSSP = 15 DELTAY OR DELSTAR = 0.0077
ITERATION 80 CIR = 0.21730 DPM = 0.00035739 AT 15 6 NSSP = 15 DELTAY OR DELSTAR = 0.0086
ITERATION 90 CIR = 0.21700 DPM = 0.00027806 AT 15 6 NSSP = 14 DELTAY OR DELSTAR = 0.0095
ITERATION 100 CIR = 0.21704 DPM = 0.00022197 AT 15 6 NSSP = 14 DELTAY OR DELSTAR = 0.0098
ITERATION 110 CIR = 0.21734 DPM = 0.00018624 AT 15 6 NSSP = 14 DELTAY OR DELSTAR = 0.0101
ITERATION 120 CIR = 0.21711 DPM = 0.00015372 AT 15 6 NSSP = 14 DELTAY OR DELSTAR = 0.0102
ITERATION 130 CIR = 0.21737 DPM = 0.00013295 AT 15 6 NSSP = 14 DELTAY OR DELSTAR = 0.0103
ITERATION 140 CIR = 0.21900 DPM = 0.00011709 AT 15 6 NSSP = 14 DELTAY OR DELSTAR = 0.0103
ITERATION 150 CIR = 0.21965 DPM = 0.00010478 AT 15 6 NSSP = 14 DELTAY OR DELSTAR = 0.0103
ITERATION 160 CIR = 0.22020 DPM = 0.00009499 AT 17 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 170 CIR = 0.22092 DPM = 0.00008461 AT 17 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 180 CIR = 0.22153 DPM = 0.00007936 AT 16 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 190 CIR = 0.22212 DPM = 0.00007396 AT 16 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 200 CIR = 0.22264 DPM = 0.00006717 AT 16 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 210 CIR = 0.22310 DPM = 0.00006223 AT 16 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 220 CIR = 0.22361 DPM = 0.00005734 AT 17 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 230 CIR = 0.22405 DPM = 0.00005305 AT 16 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 240 CIR = 0.22445 DPM = 0.00004905 AT 16 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 250 CIR = 0.22484 DPM = 0.00004453 AT 19 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 260 CIR = 0.22518 DPM = 0.00004019 AT 16 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 270 CIR = 0.22550 DPM = 0.00003562 AT 16 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 280 CIR = 0.22580 DPM = 0.00003187 AT 19 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 290 CIR = 0.22607 DPM = 0.00003392 AT 24 5 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 300 CIR = 0.22635 DPM = 0.00003052 AT 18 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 310 CIR = 0.22654 DPM = 0.00002825 AT 18 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 320 CIR = 0.22676 DPM = 0.00002611 AT 17 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 330 CIR = 0.22698 DPM = 0.00002414 AT 19 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 340 CIR = 0.22716 DPM = 0.00002217 AT 18 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 350 CIR = 0.22733 DPM = 0.00002122 AT 22 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 360 CIR = 0.22744 DPM = 0.00001997 AT 17 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 370 CIR = 0.22764 DPM = 0.00001792 AT 19 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 380 CIR = 0.22777 DPM = 0.00001609 AT 16 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 390 CIR = 0.22798 DPM = 0.00001574 AT 22 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104
ITERATION 400 CIR = 0.22801 DPM = 0.00001377 AT 19 6 NSSP = 14 DELTAY OR DELSTAR = 0.0104

## BOUNDARY LAYER ANALYSIS FOR REYNOLDS NUMBER OF 0.210E 04

X	VUDRIG	DV	SLU	VLORIG	BL	SLL
-0.460088	0.01857	0.0	0.71872	-0.01417	0.0	-0.67988
-0.42067	0.04203	0.00888	0.17769	-0.03661	0.00087	-0.17098
-0.37056	0.05153	0.00921	0.13012	-0.04629	0.00119	-0.13127
-0.339842	0.05738	0.00938	0.09058	-0.04851	0.00033	-0.05765
-0.28592	0.081140	0.00958	0.05815	-0.05176	0.00047	-0.02928
-0.18677	0.06401	0.00963	0.03398	-0.05174	0.00052	0.01488
-0.106240	0.06572	0.00988	0.01570	-0.04970	0.00078	0.04524
0.0	0.06489	0.00101	-0.02054	-0.04986	0.00122	0.10156
0.06240	0.06280	0.00116	-0.04229	-0.03260	0.00155	0.12413
0.12476	0.05513	0.00134	-0.04975	-0.02390	0.00199	0.13024
0.18677	0.03550	0.00170	-0.10375	-0.01497	0.00253	0.14060
0.24852	0.04974	0.00217	-0.13220	-0.00851	0.00343	0.11924
0.30982	0.03899	0.00291	-0.15807	-0.00450	0.00333	0.07761
0.37056	0.02459	0.00439	-0.15981	0.00363	0.00343	0.02962
0.43067	0.01250	0.00742	-0.14217	0.00468	0.00362	-0.02286
0.49000	0.00151	0.01040	-0.12734	0.00102	0.00341	-0.00948

## CP BY CENTRAL DIFFERENCES

X	CPU	CPL
-0.490	-0.083	0.389
-0.431	-0.716	-0.271
-0.371	-0.656	-0.351
-0.310	-0.715	-0.437
-0.249	-0.703	-0.421
-0.187	-0.698	-0.396
-0.125	-0.689	-0.345
-0.062	-0.702	-0.276
0.0	-0.703	-0.186
0.062	-0.657	-0.080
0.125	-0.661	0.038
0.187	-0.584	0.163
0.249	-0.380	0.273
0.310	-0.210	0.342
0.371	-0.045	0.369
0.431	-0.865	0.361
0.490	0.243	0.371
0.647	0.963	0.083
1.410	0.023	0.023
3.872	0.003	0.003

X	VU	VL	SLU	SLL
-0.49000	0.01857	-0.01417	0.71872	-0.67988
-0.43067	0.04211	-0.03468	0.17769	-0.17098
-0.37056	0.05174	-0.04648	0.13012	-0.13127
-0.339842	0.05738	-0.05174	0.09058	-0.05765
-0.28592	0.06190	-0.05224	0.05815	-0.02928
-0.18677	0.06464	-0.05236	0.03398	-0.01400
-0.12470	0.06618	-0.05049	0.01570	0.04534
-0.06240	0.06660	-0.04674	-0.00212	0.07663
0.0	0.06590	-0.04123	-0.02054	0.10156
0.06240	0.06397	-0.03416	-0.04229	0.12413
0.12470	0.06051	-0.02591	-0.04975	0.13926
0.18677	0.05521	-0.01712	-0.10175	0.14060
0.24852	0.04793	-0.00896	-0.13320	0.11936
0.30982	0.03899	-0.00285	-0.15907	0.07761
0.37056	0.02904	0.00040	-0.15981	0.02962
0.43067	0.01999	0.00666	-0.14217	-0.02284
0.49000	0.01199	-0.00240	-0.12734	-0.00948
74 74 74 74 74 75 75 75 75	74 74 74 74 74 75 75 75 75	74 74 74 74 74 75 75 75 75	74 74 74 74 74 75 75 75 75	74 74 74 74 74 75 75 75 75
74 73 73 73 73 73 73 73 74	74 73 73 73 73 73 73 73 74	74 73 73 73 73 73 73 73 74	74 73 73 73 73 73 73 73 74	74 73 73 73 73 73 73 73 74
73 71 70 69 67 66 66 66 70	73 71 70 69 67 66 66 66 70	73 71 70 69 67 66 66 66 70	73 71 70 69 67 66 66 66 70	73 71 70 69 67 66 66 66 70
73 72 68 66 65 65 65 65 68	73 72 68 66 65 65 65 65 68	73 72 68 66 65 65 65 65 68	73 72 68 66 65 65 65 65 68	73 72 68 66 65 65 65 65 68
74 75 75 75 77 77 77 77 81	74 75 75 75 77 77 77 77 81	74 75 75 75 77 77 77 77 81	74 75 75 75 77 77 77 77 81	74 75 75 75 77 77 77 77 81
74 76 79 83 87 87 87 87 91	74 76 79 83 87 87 87 87 91	74 76 79 83 87 87 87 87 91	74 76 79 83 87 87 87 87 91	74 76 79 83 87 87 87 87 91
74 78 82 86 92 92 92 92 98	74 78 82 86 92 92 92 92 98	74 78 82 86 92 92 92 92 98	74 78 82 86 92 92 92 92 98	74 78 82 86 92 92 92 92 98
74 79 83 87 91 91 91 91 98	74 79 83 87 91 91 91 91 98	74 79 83 87 91 91 91 91 98	74 79 83 87 91 91 91 91 98	74 79 83 87 91 91 91 91 98
74 78 82 85 88 88 88 88 93	74 78 82 85 88 88 88 88 93	74 78 82 85 88 88 88 88 93	74 78 82 85 88 88 88 88 93	74 78 82 85 88 88 88 88 93
74 77 80 83 85 85 85 85 93	74 77 80 83 85 85 85 85 93	74 77 80 83 85 85 85 85 93	74 77 80 83 85 85 85 85 93	74 77 80 83 85 85 85 85 93
74 76 78 80 82 82 82 82 93	74 76 78 80 82 82 82 82 93	74 76 78 80 82 82 82 82 93	74 76 78 80 82 82 82 82 93	74 76 78 80 82 82 82 82 93
74 75 76 77 78 78 78 78 91	74 75 76 77 78 78 78 78 91	74 75 76 77 78 78 78 78 91	74 75 76 77 78 78 78 78 91	74 75 76 77 78 78 78 78 91
74 74 73 73 73 73 73 73 84	74 74 73 73 73 73 73 73 84	74 74 73 73 73 73 73 73 84	74 74 73 73 73 73 73 73 84	74 74 73 73 73 73 73 73 84
73 72 71 70 69 68 68 68 83	73 72 71 70 69 68 68 68 83	73 72 71 70 69 68 68 68 83	73 72 71 70 69 68 68 68 83	73 72 71 70 69 68 68 68 83
73 71 69 68 67 66 66 66 82	73 71 69 68 67 66 66 66 82	73 71 69 68 67 66 66 66 82	73 71 69 68 67 66 66 66 82	73 71 69 68 67 66 66 66 82
73 70 68 66 65 65 65 65 80	73 70 68 66 65 65 65 65 80	73 70 68 66 65 65 65 65 80	73 70 68 66 65 65 65 65 80	73 70 68 66 65 65 65 65 80
72 70 67 65 65 65 65 65 79	72 70 67 65 65 65 65 65 79	72 70 67 65 65 65 65 65 79	72 70 67 65 65 65 65 65 79	72 70 67 65 65 65 65 65 79
71 69 68 67 66 66 66 66 78	71 69 68 67 66 66 66 66 78	71 69 68 67 66 66 66 66 78	71 69 68 67 66 66 66 66 78	71 69 68 67 66 66 66 66 78
72 71 71 71 71 71 71 71 75	72 71 71 71 71 71 71 71 75	72 71 71 71 71 71 71 71 75	72 71 71 71 71 71 71 71 75	72 71 71 71 71 71 71 71 75
74 76 74 74 74 74 74 74 75	74 76 74 74 74 74 74 74 75	74 76 74 74 74 74 74 74 75	74 76 74 74 74 74 74 74 75	74 76 74 74 74 74 74 74 75
74 75 75 75 75 75 75 75 75	74 75 75 75 75 75 75 75 75	74 75 75 75 75 75 75 75 75	74 75 75 75 75 75 75 75 75	74 75 75 75 75 75 75 75 75

WAVE CD = 0.003733

## CHART 100

-0.4900		U	T	B		L					
-0.4300	U		L		T	B					
-0.3700	U		L		T	B					
-0.2900	U		L		T	B					
-0.2300	U		L		T	B					
-0.1700	U		L		T	B					
-0.1100	U		L		T	B					
-0.0500	U		L		T	B					
0.0100	U		L	T	B						
0.0700	U			LT	B						
0.1300	U			T	BL						
0.1900	U			T	B	L					
0.2500		U		T	B	L					
0.3100		U		T	B	L					
0.3900				UT	B	L					
0.4500				T	B	U					
0.4900				TB		U	L				
PRESSURE COEFFICIENT											
-0.716	-0.604	-0.493	-0.382	-0.270	-0.159	-0.047	0.064	0.176	0.287	0.399	
CPSTAR = -0.5842 CLCIR = 0.4560											
CL = 0.4384 CD = 0.010745 CMLE = -0.2306 CDF = 0.007012 CMCA = -0.1210											

SAMPLE CASE NO. 2--KORN AIRFOIL 75-06-12 TWO GRID HALVINGS ONLY

K-Y GRID SYSTEM

```

2 -0.0027E 01   3 -0.3872E 01   4 -0.2276E 01   5 -0.1410E 01   6 -0.915E 00   7 -0.6471E 00
8 -0.5317E 00   9 -0.4900E 00   10 -0.4604E 00   11 -0.4307E 00   12 -0.4007E 00   13 -0.3706E 00
14 -0.3403E 00   15 -0.3098E 00   16 -0.2792E 00   17 -0.2485E 00   18 -0.2177E 00   19 -0.1868E 00
20 -0.1558E 00   21 -0.1247E 00   22 -0.0935E-01   23 -0.0624E-01   24 -0.0312E-01   25 0.0
26 0.0217E-01   27 0.0624E-01   28 0.0935E-01   29 0.1247E 00   30 0.1558E 00   31 0.1868E 00
32 0.2177E 00   33 0.2485E 00   34 0.2792E 00   35 0.3098E 00   36 0.3403E 00   37 0.3706E 00
38 0.4007E 00   39 0.4307E 00   40 0.4604E 00   41 0.4900E 00   42 0.5317E 00   43 0.6471E 00
44 0.915E 00   45 0.1410E 01   46 0.2276E 01   47 0.3872E 01   48 0.5317E 01
2 -0.1669E 01   3 -0.9181E 00   4 -0.5939E 00   5 -0.3403E 00   6 -0.3206E 00   7 -0.2460E 00
8 -0.1868E 00   9 -0.1420E 00   10 -0.1010E 00   11 -0.6592E-01   12 -0.3239E-01   13 -0.9213E-07
14 0.3239E-01   15 0.6592E-01   16 0.1010E 00   17 0.1420E 00   18 0.1688E 00   19 0.2460E 00
20 0.3206E 00   21 0.4261E 00   22 0.5939E 00   23 0.9181E 00   24 0.1669E 01

```

MACH NO. IS 0.752 ANGLE OF ATTACK IS 0.0 DEGREES  
DIRECT SOLUTION TO 0.50  
CASE NUMBER 100

INVISCID ANALYSIS CASE  
WITH VISCVLS INTERACTION

```

CFINP
M= 0.75199997   ,W= 1.09999998   ,X1= 0.50000000   ,X2= 10000.000   ,ALPX= 0.0   ,EPS= 0.0   ,EPSS=
0.39999998   ,X4= 0.48999995   ,S2= 0.00000000   ,CONV= 0.99999997E-05,A1= 0.24500000   ,A2= 0.14999998   ,A3= 3.8699999
RN= 20500000.   ,XIBDLY= -0.44000000   ,CIR= 0.22801352   ,CDCORR= 0.0   ,RDEL= 0.25000000   ,RDelfN= 0.12500000   ,
SP= 0.39999969E-02,XSEP= 0.44000000   ,RCPB= 0.19999999   ,CPB= 0.39999998   ,XMN= 0.46999997   ,XLSEP= 0.50000000   ,XPC=
0.99999964E-01
END
&INP
IMAX= 49,JMAX= 25,IKASE= 100,INV= 0,MITER= 200,NHALF= 2,ITACT= 1,ISKP2=
0,ISKP3= 0,ISKP4= 0,ITERP= 0,IREAD= 0,LP= 1000,ITEUPC= 0,TELWC=
0
END

AIRFOIL COORDINATES
X      YU      YL      UPPER SLOPE LOWER SLOPE

```

```

-0.49000  0.01657  -0.01417  0.71872  -0.67988
-0.46044  0.03318  -0.02691  0.36638  -0.31117
-0.43087  0.04211  -0.03468  0.17769  -0.17098
-0.40072  0.04692  -0.03958  0.15391  -0.15113
-0.37058  0.05174  -0.04448  0.13012  -0.13127
-0.34028  0.05474  -0.04716  0.10430  -0.09446
-0.30982  0.05774  -0.04983  0.07848  -0.05765
-0.28936  0.06174  -0.05251  0.06753  -0.03893
-0.26852  0.065190  -0.05524  0.05615  -0.02020
-0.21769  0.06327  -0.05230  0.05086  -0.01700
-0.18677  0.06464  -0.05236  0.03398  0.01480
-0.15577  0.05451  -0.05142  0.02464  0.03007
-0.12470  0.06618  -0.05049  0.01570  0.04534
-0.09357  0.06638  -0.04861  0.00679  0.05994
-0.06240  0.06660  -0.04674  -0.00212  0.07463
-0.03121  0.06625  -0.04390  -0.01133  0.08809
0.0  0.06590  -0.04123  -0.02054  0.10156
0.03121  0.06494  -0.03763  -0.03142  0.11284
0.06240  0.06397  -0.03416  -0.04229  0.12413
0.09357  0.06224  -0.03003  -0.05602  0.13169
0.12470  0.06051  -0.02591  -0.06975  0.13926
0.15577  0.05786  -0.02152  -0.08575  0.13993
0.18677  0.05520  -0.01712  -0.10175  0.14060
0.21769  0.05157  -0.01378  -0.11738  0.14166
0.24852  0.04793  -0.00986  -0.13320  0.14236
0.27933  0.04344  -0.00590  -0.14614  0.14994
0.30082  0.03894  -0.00285  -0.15907  0.07761
0.34028  0.03399  -0.00122  -0.15944  0.05362
0.37058  0.02904  0.00040  -0.15981  0.02962
0.40072  0.02451  0.00053  -0.15094  0.00339
0.43067  0.01999  0.00066  -0.14217  -0.02284
0.46044  0.01599  -0.00087  -0.13475  -0.05166
0.49000  0.01199  -0.00240  -0.12733  -0.08048
ITERATION 10 CIR = 0.22783 DPM = 0.000033033 AT 32 7 NSSP = 57 DELTAY OR DELSTAR = 0.0104
ITERATION 20 CIR = 0.22797 DPM = 0.00007684 AT 24 24 NSSP = 58 DELTAY OR DELSTAR = 0.0104
ITERATION 30 CIR = 0.22812 DPM = 0.00004059 AT 37 2 NSSP = 58 DELTAY OR DELSTAR = 0.0104
ITERATION 40 CIR = 0.22824 DPM = 0.00002819 AT 35 2 NSSP = 58 DELTAY OR DELSTAR = 0.0104
ITERATION 50 CIR = 0.22834 DPM = 0.00002676 AT 46 12 NSSP = 57 DELTAY OR DELSTAR = 0.0104
ITERATION 60 CIR = 0.22724 DPM = 0.000029546 AT 12 15 NSSP = 60 DELTAY OR DELSTAR = 0.0101
ITERATION 70 CIR = 0.22756 DPM = 0.000009030 AT 14 15 NSSP = 62 DELTAY OR DELSTAR = 0.0094
ITERATION 80 CIR = 0.22719 DPM = 0.00004506 AT 14 15 NSSP = 62 DELTAY OR DELSTAR = 0.0095
ITERATION 90 CIR = 0.22771 DPM = 0.00001532 AT 48 12 NSSP = 62 DELTAY OR DELSTAR = 0.0094
ITERATION 100 CIR = 0.22785 DPM = 0.00002313 AT 15 15 NSSP = 62 DELTAY OR DELSTAR = 0.0091
ITERATION 110 CIR = 0.22768 DPM = 0.00001861 AT 47 13 NSSP = 62 DELTAY OR DELSTAR = 0.0092
ITERATION 120 CIR = 0.22799 DPM = 0.00002056 AT 47 13 NSSP = 62 DELTAY OR DELSTAR = 0.0090
ITERATION 130 CIR = 0.22788 DPM = 0.00002754 AT 48 11 NSSP = 62 DELTAY OR DELSTAR = 0.0092
ITERATION 140 CIR = 0.22771 DPM = 0.00001436 AT 44 13 NSSP = 62 DELTAY OR DELSTAR = 0.0093
ITERATION 150 CIR = 0.22755 DPM = 0.00001826 AT 48 13 NSSP = 62 DELTAY OR DELSTAR = 0.0093
ITERATION 160 CIR = 0.22741 DPM = 0.00001532 AT 48 12 NSSP = 62 DELTAY OR DELSTAR = 0.0094
ITERATION 170 CIR = 0.22726 DPM = 0.00001511 AT 46 13 NSSP = 62 DELTAY OR DELSTAR = 0.0094
ITERATION 180 CIR = 0.22731 DPM = 0.00001040 AT 31 15 NSSP = 62 DELTAY OR DELSTAR = 0.0093
ITERATION 190 CIR = 0.22720 DPM = 0.00001144 AT 42 13 NSSP = 62 DELTAY OR DELSTAR = 0.0094
ITERATION 200 CIR = 0.22709 DPM = 0.00001460 AT 47 7 NSSP = 62 DELTAY OR DELSTAR = 0.0094

```

BOUNDARY LAYER ANALYSIS FOR REYNOLDS NUMBER OF 0.210E 08

X	YUDRIG	DU	SLU	YLDRIG	DL	SLL
-0.49000	0.01657	0.0	0.71872	-0.01417	0.0	-0.47988
-0.46044	0.03318	0.00001	0.36638	-0.02691	0.00001	-0.31117
-0.43067	0.04203	0.00005	0.24494	-0.03461	0.00005	-0.22472
-0.40072	0.04758	0.00012	0.15056	-0.04017	0.00011	-0.15910
-0.37058	0.05153	0.00021	0.11970	-0.04429	0.00019	-0.12047
-0.34028	0.05473	0.00030	0.09825	-0.04736	0.00026	-0.08720
-0.30982	0.05738	0.00038	0.08150	-0.04951	0.00033	-0.06020
-0.27923	0.05958	0.00046	0.06783	-0.05095	0.00043	-0.03884
-0.24852	0.06140	0.00053	0.05565	-0.05176	0.00046	-0.01937
-0.21769	0.06286	0.00061	0.04433	-0.05201	0.00052	-0.00181
-0.18677	0.06401	0.00068	0.03448	-0.05174	0.00062	0.01457
-0.15577	0.06486	0.00074	0.02471	-0.05096	0.00070	0.03026
-0.12470	0.06542	0.00080	0.01538	-0.04970	0.00079	0.04542
-0.09357	0.06571	0.00086	0.00668	-0.04797	0.00088	0.06022
-0.06240	0.06572	0.00092	-0.00227	-0.04576	0.00098	0.07468
-0.03121	0.06545	0.00097	-0.01127	-0.04310	0.00109	0.08857
0.0	0.06489	0.00103	-0.02074	-0.04000	0.00122	0.10139
0.03121	0.06402	0.00110	-0.03105	-0.03649	0.00137	0.11360
0.06240	0.06280	0.00118	-0.04234	-0.03260	0.00154	0.12447
0.09357	0.06119	0.00127	-0.05515	-0.02838	0.00174	0.13299
0.12470	0.05913	0.00139	-0.06954	-0.02390	0.00198	0.13908
0.15577	0.05658	0.00154	-0.08554	-0.01925	0.00225	0.14153
0.18677	0.05350	0.00171	-0.10202	-0.01457	0.00256	0.13927
0.21769	0.04989	0.00191	-0.11823	-0.01006	0.00286	0.13190
0.24852	0.04574	0.00217	-0.13341	-0.00591	0.00313	0.11894
0.27923	0.04110	0.00249	-0.14692	-0.00234	0.00332	0.10064
0.30982	0.03599	0.00292	-0.15846	0.00050	0.00341	0.07903
0.34028	0.03045	0.00349	-0.16610	0.00256	0.00343	0.05558
0.37058	0.02459	0.00424	-0.16965	0.00383	0.00338	0.03145
0.40072	0.01855	0.00514	-0.17077	0.00434	0.00331	0.00710
0.43067	0.01250	0.00623	-0.15510	0.00408	0.00319	-0.01580
0.46044	0.00672	0.00779	-0.13173	0.00304	0.00301	-0.04429
0.49000	0.00151	0.00933	-0.11632	0.00102	0.00282	-0.08015

CP BY CENTRAL DIFFERENCES

X	CPU	CPL
-0.490	0.157	0.443
-0.460	-0.367	-0.059
-0.431	-0.733	-0.229
-0.401	-0.854	-0.345
-0.371	-0.840	-0.400
-0.340	-0.824	-0.439
-0.310	-0.800	-0.450
-0.279	-0.774	-0.449
-0.249	-0.750	-0.443
-0.218	-0.726	-0.429
-0.187	-0.700	-0.410
-0.156	-0.702	-0.389
-0.125	-0.701	-0.356
-0.094	-0.706	-0.322
-0.062	-0.713	-0.282
-0.031	-0.721	-0.237
0.0	-0.724	-0.180
0.031	-0.723	-0.135
0.062	-0.719	-0.074
0.094	-0.699	-0.014
0.125	-0.655	0.044
0.156	-0.595	0.105
0.187	-0.529	0.164
0.218	-0.457	0.220
0.249	-0.380	0.268
0.279	-0.300	0.327
0.310	-0.215	0.335
0.340	-0.126	0.352
0.371	-0.045	0.361
0.401	0.035	0.360
0.431	0.113	0.354
0.460	0.154	0.343
0.490	0.213	0.310
0.532	0.181	0.181
0.647	0.057	0.057
0.912	0.047	0.047
1.410	0.020	0.020
2.276	0.008	0.008
3.872	0.003	0.003
6.027	0.001	0.001

WAVE CD = 0.006996

## CHART 100

-0.4900					T	B						
-0.4500												
-0.4300	U		U	L	L	T	B		U		L	
-0.3900	U	U		L		T	B					
-0.3700	U			L		T	B					
-0.3300	U			L		T	B					
-0.2900	U	U		L		T	B					
-0.2700	U			L		T	B					
-0.2300	U	U		L		T	B					
-0.2100	U			L		T	B					
-0.1700	U	U		L		T	B					
-0.1500	U			L		T	B					
-0.1100	U	U		L		T	B					
-0.0900	U			L		T	B					
-0.0500	U	U		L	L	T	B					
-0.0300	U			L	L	T	B					
0.0100	U			L		T	B					
0.0500	U	U			L	T	B					
0.0700	U				L	T	B					
0.1100	U	U				T	L	B				
0.1300						T	B	L				
0.1700	U	U				T	B		L			
0.1900						T	B		L			
0.2300	U	U				T	B		L			
0.2500						T	B		L			
0.2900	U		U			T	B		L			
0.3100			U	U		T	B		L			
0.3500					U	T	B		L			
0.3900						U	T	B	L			
0.4100						T	B	U	L			
0.4500						T	B	U	L			
0.4700						T	B	U	L			
0.4900						T	B	U	L			
.	.	.	.	.	.	.	.	.	.	.	.	
-0.654	-0.724	-0.595	-0.465	-0.335	-0.205	-0.076	0.054	0.184	0.314	0.443		
PRESSURE COEFFICIENT												
CPSTAR = -0.5842				CLCIR = 0.4542								
CL = 0.4481 CD = 0.014111 CMLE = -0.2323 CDF = 0.007115 CM4 = -0.1203												

## REFERENCES

1. Carlson, L.A.: "Transonic Airfoil Flowfield Analysis Using Cartesion Coordinates," NASA CR-2577, August 1975.
2. Carlson, L.A.: "Transonic Airfoil Design Using Cartesion Coordinates," NASA CR-2578, April 1976.
3. Carlson, L.A.: "Transonic Airfoil Analysis and Design Using Cartesian Coordinates," AIAA Journal, Vol. 13, No. 5, May 1976, pp. 349-356.
4. South, J.C., Jr. and Jameson, A.: "Relaxation Solutions for Inviscid Axisymmetric Transonic Flow over Blunt or Pointed Bodies," Proc. AIAA Computational Fluid Dynamics Conf., Palm Springs, Calif., July 1973, pp. 8-17.
5. Nash, J.F. and Macdonald, A.G.J.: "The Calculation of Momentum Thickness in a Turbulent Boundary Layer at Mach Numbers up to Unity," Aero. Res. Council C.P. No. 963, 1967.
6. Bauer, F.; Garabedian, P.; Korn, D.; and Jameson, A.: "Supercritical Wing Sections II," Springer-Verlag, New York, 1975.
7. Bavitz, P.C.: "An Analysis Method for Two-Dimensional Transonic Viscous Flow," NASA TN-D-7718, January 1975.